



(51) International Patent Classification:

H04N 19/30 (2014.01) H04N 19/00 (2014.01)

H04N 19/70 (2014.01)

(21) International Application Number:

PCT/CN2022/084992

(22) International Filing Date:

02 April 2022 (02.04.2022)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2021/085292

02 April 2021 (02.04.2021) CN

(71) Applicants: BEIJING BYTEDANCE NETWORK

TECHNOLOGY CO., LTD. [CN/CN]; Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District, Beijing 100041 (CN). BYTEDANCE INC. [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) Inventors: WANG, Yang; Jinritoutiao Post Office, Chi-

na Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). WANG, Ye-Kui; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). ZHANG, Li;

12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(74) Agent: LIU, SHEN & ASSOCIATES; 10th Floor, Building 1, 10 Caihefang Road, Haidian District, Beijing 100080 (CN).

(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, IT, JM, JO, JP, KE, KG, KH, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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,

(54) Title: SCALABILITY DIMENSION INFORMATION IN VIDEO CODING

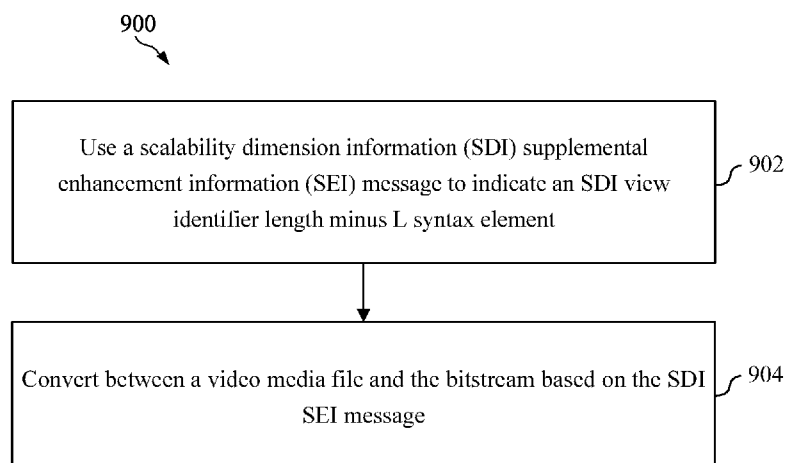


FIG. 9

(57) Abstract: A method of processing video data includes using a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and converting between a video media file and the bitstream based on the SDI SEI message. A corresponding video coding apparatus and non-transitory computer readable medium are also disclosed.

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,  
KM, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17:**

— *of inventorship (Rule 4.17(iv))*

**Published:**

— *with international search report (Art. 21(3))*

## SCALABILITY DIMENSION INFORMATION IN VIDEO CODING

### CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This patent application claims the benefit of International Application No. PCT/CN2021/085292 filed on April 2, 2021, by Beijing ByteDance Network Technology Co., Ltd., and titled “Scalability Dimension Information Improvements,” which is hereby incorporated by reference.

### TECHNICAL FIELD

[0002] The present disclosure is generally related to video coding and, in particular, to supplemental enhancement information (SEI) messages used in image/video coding.

### BACKGROUND

[0003] Digital video accounts for the largest bandwidth use on the internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, it is expected that the bandwidth demand for digital video usage will continue to grow.

### SUMMARY

[0004] The disclosed aspects/embodiments provide techniques that use a scalability dimension information (SDI) view identifier length minus L syntax element to prevent a length of an SDI view ID value syntax element, which specifies a view identifier of an i-th layer in a bitstream, from being zero. The disclosed aspects/embodiments further provide techniques that prevent a bitstream from having a multiview acquisition information supplemental enhancement information (SEI) message or an auxiliary information SEI message when an SDI message is not present in the bitstream. The disclosed aspects/embodiments also provide techniques that prevent the multiview acquisition information SEI message from being scalable-nested.

[0005] A first aspect relates to a method of processing video data. The method includes using a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and performing a conversion between a video media file and the bitstream based on the SDI SEI message.

[0006] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the SDI view identifier length minus L syntax element is configured to prevent a

length of an SDI view identifier value syntax element, which specifies a view identifier of an  $i$ -th layer in a bitstream, from being zero.

[0007] Optionally, in any of the preceding aspects, another implementation of the aspect provides that  $L$  is equal to 1.

[0008] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the SDI view identifier length minus  $L$  syntax element is designated `sdi_view_id_len_minus1`.

[0009] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the SDI view identifier value syntax element is designated `sdi_view_id_val[ i ]`.

[0010] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the SDI view identifier length minus  $L$  syntax element, plus one, specifies the length of the SDI view identifier value syntax element.

[0011] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the SDI view identifier length minus  $L$  syntax element is coded as an unsigned integer using  $N$  bits.

[0012] Optionally, in any of the preceding aspects, another implementation of the aspect provides that  $N$  is equal to 4.

[0013] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the SDI view identifier length minus  $L$  syntax element is coded as a fixed-pattern bitstring using  $N$  bits, a signed integer using  $N$  bits, a truncated binary, a signed integer  $K$ -th order Exp-Golomb-coded syntax element where  $K$  is equal to 0, or an unsigned integer  $M$ -th order Exp-Golomb-coded syntax element where  $M$  is equal to 0.

[0014] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the bitstream is a bitstream in scope.

[0015] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a multiview information SEI message and an auxiliary information SEI message are not present in a coded video sequence (CVS) unless the SDI SEI message is present in the CVS.

[0016] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the multiview information SEI message comprises a multiview acquisition information SEI message.

[0017] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the auxiliary information SEI message comprises a depth representation information SEI message.

[0018] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the auxiliary information SEI message comprises an alpha channel information SEI message.

[0019] Optionally, in any of the preceding aspects, another implementation of the aspect provides that one or more of an SDI multiview information flag and an SDI auxiliary information flag are equal to 1 when the multiview information SEI message or the auxiliary information SEI message are present in the bitstream.

[0020] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the multiview information SEI message comprises a multiview acquisition information SEI message, and wherein the multiview acquisition information SEI message is not scalable-nested.

[0021] Optionally, in any of the preceding aspects, another implementation of the aspect provides that an SEI message in the bitstream and having a payload type equal to 179 is constrained from being included in a scalable nesting SEI message.

[0022] Optionally, in any of the preceding aspects, another implementation of the aspect provides that an SEI message in the bitstream and having a payload type equal to 3, 133, 179, 180, or 205 is constrained from being included in a scalable nesting SEI message.

[0023] A second aspect relates to an apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor cause the processor to: use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and convert between a video media file and the bitstream based on the SDI SEI message.

[0024] A third aspect relates to a non-transitory computer readable medium comprising a computer program product for use by a coding apparatus, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium that, when executed by one or more processors, cause the coding apparatus to: use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and convert between a video media file and the bitstream based on the SDI SEI message.

[0025] A fourth aspect relates to a non-transitory computer-readable storage medium storing instructions that cause a processor to: use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and convert between a video media file and the bitstream based on the SDI SEI message.

[0026] A fifth aspect relates to a non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises: use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and convert between a video media file and the bitstream based on the SDI SEI message.

[0027] A sixth aspect relates to a method for storing bitstream of a video, comprising: using a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; generating the bitstream based on the SDI SEI message; and storing the bitstream in a non-transitory computer-readable recording medium.

[0028] For the purpose of clarity, any one of the foregoing embodiments may be combined with any one or more of the other foregoing embodiments to create a new embodiment within the scope of the present disclosure.

[0029] These and other features will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings and claims.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0030] For a more complete understanding of this disclosure, reference is now made to the following brief description, taken in connection with the accompanying drawings and detailed description, wherein like reference numerals represent like parts.

[0031] FIG. 1 illustrates an example of multi-layer coding for spatial scalability.

[0032] FIG. 2 illustrates an example of multi-layer coding using output layer sets (OLSs).

[0033] FIG. 3 illustrates an embodiment of a video bitstream.

[0034] FIG. 4 is a block diagram showing an example video processing system.

[0035] FIG. 5 is a block diagram of a video processing apparatus.

[0036] FIG. 6 is a block diagram that illustrates an example video coding system.

[0037] FIG. 7 is a block diagram illustrating an example of video encoder.

[0038] FIG. 8 is a block diagram illustrating an example of video decoder.

[0039] FIG. 9 is a method for coding video data according to an embodiment of the disclosure.

### **DETAILED DESCRIPTION**

[0040] It should be understood at the outset that although an illustrative implementation of one or more embodiments are provided below, the disclosed systems and/or methods may be

implemented using any number of techniques, whether currently known or in existence. The disclosure should in no way be limited to the illustrative implementations, drawings, and techniques illustrated below, including the exemplary designs and implementations illustrated and described herein, but may be modified within the scope of the appended claims along with their full scope of equivalents.

**[0041]** Video coding standards have evolved primarily through the development of the well-known International Telecommunication Union - Telecommunication (ITU-T) and International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) standards. The ITU-T produced H.261 and H.263, ISO/IEC produced Moving Picture Experts Group (MPEG)-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/High Efficiency Video Coding (HEVC) standards. See ITU-T and ISO/IEC, “High efficiency video coding”, Rec. ITU-T H.265 | ISO/IEC 23008-2 (in force edition). Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, the Joint Video Exploration Team (JVET) was founded by Video Coding Experts Group (VCEG) and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM). See J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, “Algorithm description of Joint Exploration Test Model 7 (JEM7),” JVET-G1001, Aug. 2017. The JVET was later renamed to be the Joint Video Experts Team (JVET) when the Versatile Video Coding (VVC) project officially started. VVC is the new coding standard, targeting at 50% bitrate reduction as compared to HEVC, that has been finalized by the JVET at its 19th meeting ended at July 1, 2020. See Rec. ITU-T H.266 | ISO/IEC 23090-3, “Versatile Video Coding”, 2020.

**[0042]** The VVC standard (ITU-T H.266 | ISO/IEC 23090-3) and the associated Versatile Supplemental Enhancement Information (VSEI) standard (ITU-T H.274 | ISO/IEC 23002-7) have been designed for use in a maximally broad range of applications, including both the traditional uses such as television broadcast, video conferencing, or playback from storage media, and also newer and more advanced use cases such as adaptive bit rate streaming, video region extraction, composition and merging of content from multiple coded video bitstreams, multiview video, scalable layered coding, and viewport-adaptive 360° immersive media. See B. Bross, J. Chen, S. Liu, Y.-K. Wang (editors), “Versatile Video Coding (Draft 10),” JVET-S2001, Rec. ITU-T Rec. H.274 | ISO/IEC 23002-7, “Versatile Supplemental Enhancement Information Messages for Coded Video Bitstreams”, 2020, and J. Boyce, V. Drugeon, G. Sullivan, Y.-K.

Wang (editors), “Versatile supplemental enhancement information messages for coded video bitstreams (Draft 5),” JVET-S2007.

[0043] The Essential Video Coding (EVC) standard (ISO/IEC 23094-1) is another video coding standard that has recently been developed by MPEG.

[0044] FIG. 1 is a schematic diagram illustrating an example of layer based prediction 100. Layer based prediction 100 is compatible with unidirectional inter-prediction and/or bidirectional inter-prediction, but is also performed between pictures in different layers.

[0045] Layer based prediction 100 is applied between pictures 111, 112, 113, and 114 and pictures 115, 116, 117, and 118 in different layers. In the example shown, pictures 111, 112, 113, and 114 are part of layer N+1 132 and pictures 115, 116, 117, and 118 are part of layer N 131. A layer, such as layer N 131 and/or layer N+1 132, is a group of pictures that are all associated with a similar value of a characteristic, such as a similar size, quality, resolution, signal to noise ratio, capability, etc. In the example shown, layer N+1 132 is associated with a larger image size than layer N 131. Accordingly, pictures 111, 112, 113, and 114 in layer N+1 132 have a larger picture size (e.g., larger height and width and hence more samples) than pictures 115, 116, 117, and 118 in layer N 131 in this example. However, such pictures can be separated between layer N+1 132 and layer N 131 by other characteristics. While only two layers, layer N+1 132 and layer N 131, are shown, a set of pictures can be separated into any number of layers based on associated characteristics. Layer N+1 132 and layer N 131 may also be denoted by a layer ID. A layer ID is an item of data that is associated with a picture and denotes the picture is part of an indicated layer. Accordingly, each picture 111-118 may be associated with a corresponding layer ID to indicate which layer N+1 132 or layer N 131 includes the corresponding picture.

[0046] Pictures 111-118 in different layers 131-132 are configured to be displayed in the alternative. As such, pictures 111-118 in different layers 131-132 can share the same temporal identifier (ID) and can be included in the same access unit (AU) 106. As used herein, an AU is a set of one or more coded pictures associated with the same display time for output from a decoded picture buffer (DPB). For example, a decoder may decode and display picture 115 at a current display time if a smaller picture is desired or the decoder may decode and display picture 111 at the current display time if a larger picture is desired. As such, pictures 111-114 at higher layer N+1 132 contain substantially the same image data as corresponding pictures 115-118 at lower layer N 131 (notwithstanding the difference in picture size). Specifically, picture 111 contains substantially the same image data as picture 115, picture 112 contains substantially the same image data as picture 116, etc.

**[0047]** Pictures 111-118 can be coded by reference to other pictures 111-118 in the same layer N 131 or N+1 132. Coding a picture in reference to another picture in the same layer results in inter-prediction 123, which is compatible unidirectional inter-prediction and/or bidirectional inter-prediction. Inter-prediction 123 is depicted by solid line arrows. For example, picture 113 may be coded by employing inter-prediction 123 using one or two of pictures 111, 112, and/or 114 in layer N+1 132 as a reference, where one picture is referenced for unidirectional inter-prediction and/or two pictures are referenced for bidirectional inter-prediction. Further, picture 117 may be coded by employing inter-prediction 123 using one or two of pictures 115, 116, and/or 118 in layer N 131 as a reference, where one picture is referenced for unidirectional inter-prediction and/or two pictures are referenced for bidirectional inter-prediction. When a picture is used as a reference for another picture in the same layer when performing inter-prediction 123, the picture may be referred to as a reference picture. For example, picture 112 may be a reference picture used to code picture 113 according to inter-prediction 123. Inter-prediction 123 can also be referred to as intra-layer prediction in a multi-layer context. As such, inter-prediction 123 is a mechanism of coding samples of a current picture by reference to indicated samples in a reference picture that are different from the current picture where the reference picture and the current picture are in the same layer.

**[0048]** Pictures 111-118 can also be coded by reference to other pictures 111-118 in different layers. This process is known as inter-layer prediction 121, and is depicted by dashed arrows. Inter-layer prediction 121 is a mechanism of coding samples of a current picture by reference to indicated samples in a reference picture where the current picture and the reference picture are in different layers and hence have different layer IDs. For example, a picture in a lower layer N 131 can be used as a reference picture to code a corresponding picture at a higher layer N+1 132. As a specific example, picture 111 can be coded by reference to picture 115 according to inter-layer prediction 121. In such a case, the picture 115 is used as an inter-layer reference picture. An inter-layer reference picture is a reference picture used for inter-layer prediction 121. In most cases, inter-layer prediction 121 is constrained such that a current picture, such as picture 111, can only use inter-layer reference picture(s) that are included in the same AU 106 and that are at a lower layer, such as picture 115. When multiple layers (e.g., more than two) are available, inter-layer prediction 121 can encode/decode a current picture based on multiple inter-layer reference picture(s) at lower levels than the current picture.

**[0049]** A video encoder can employ layer based prediction 100 to encode pictures 111-118 via many different combinations and/or permutations of inter-prediction 123 and inter-layer prediction 121. For example, picture 115 may be coded according to intra-prediction. Pictures

116-118 can then be coded according to inter-prediction 123 by using picture 115 as a reference picture. Further, picture 111 may be coded according to inter-layer prediction 121 by using picture 115 as an inter-layer reference picture. Pictures 112-114 can then be coded according to inter-prediction 123 by using picture 111 as a reference picture. As such, a reference picture can serve as both a single layer reference picture and an inter-layer reference picture for different coding mechanisms. By coding higher layer N+1 132 pictures based on lower layer N 131 pictures, the higher layer N+1 132 can avoid employing intra-prediction, which has much lower coding efficiency than inter-prediction 123 and inter-layer prediction 121. As such, the poor coding efficiency of intra-prediction can be limited to the smallest/lowest quality pictures, and hence limited to coding the smallest amount of video data. The pictures used as reference pictures and/or inter-layer reference pictures can be indicated in entries of reference picture list(s) contained in a reference picture list structure.

**[0050]** Each AU 106 in FIG. 1 may contain several pictures. For example, one AU 106 may contain pictures 111 and 115. Another AU 106 may contain pictures 112 and 116. Indeed, each AU 106 is a set of one or more coded pictures associated with the same display time (e.g., the same temporal ID) for output from a decoded picture buffer (DPB) (e.g., for display to a user). Each access unit delimiter (AUD) 108 is an indicator or data structure used to indicate the start of an AU (e.g., AU 108) or the boundary between AUs.

**[0051]** Previous H.26x video coding families have provided support for scalability in separate profile(s) from the profile(s) for single-layer coding. Scalable video coding (SVC) is the scalable extension of the AVC/H.264 that provides support for spatial, temporal, and quality scalabilities. For SVC, a flag is signaled in each macroblock (MB) in enhancement layer (EL) pictures to indicate whether the EL MB is predicted using the collocated block from a lower layer. The prediction from the collocated block may include texture, motion vectors, and/or coding modes. Implementations of SVC cannot directly reuse unmodified H.264/AVC implementations in their design. The SVC EL macroblock syntax and decoding process differs from H.264/AVC syntax and decoding process.

**[0052]** Scalable HEVC (SHVC) is the extension of the HEVC/H.265 standard that provides support for spatial and quality scalabilities, multiview HEVC (MV-HEVC) is the extension of the HEVC/H.265 that provides support for multi-view scalability, and 3D HEVC (3D-HEVC) is the extension of the HEVC/H.264 that provides support for three dimensional (3D) video coding that is more advanced and more efficient than MV-HEVC. Note that the temporal scalability is included as an integral part of the single-layer HEVC codec. The design of the multi-layer extension of HEVC employs the idea where the decoded pictures used for inter-layer prediction

come only from the same AU and are treated as long-term reference pictures (LTRPs), and are assigned reference indices in the reference picture list(s) along with other temporal reference pictures in the current layer. Inter-layer prediction (ILP) is achieved at the prediction unit (PU) level by setting the value of the reference index to refer to the inter-layer reference picture(s) in the reference picture list(s).

**[0053]** Notably, both reference picture resampling and spatial scalability features call for resampling of a reference picture or part thereof. Reference picture resampling (RPR) can be realized at either the picture level or coding block level. However, when RPR is referred to as a coding feature, it is a feature for single-layer coding. Even so, it is possible or even preferable from a codec design point of view to use the same resampling filter for both the RPR feature of single-layer coding and the spatial scalability feature for multi-layer coding.

**[0054]** FIG. 2 illustrates an example of layer based prediction 200 utilizing output layer sets (OLSs). Layer based prediction 100 is compatible with unidirectional inter-prediction and/or bidirectional inter-prediction, but is also performed between pictures in different layers. The layer based prediction of FIG. 2 is similar to that of FIG. 1. Therefore, for the sake of brevity, a full description of layer based prediction is not repeated.

**[0055]** Some of the layers in the coded video sequence (CVS) 290 of FIG. 2 are included in an OLS. An OLS is a set of layers for which one or more layers are specified as the output layers. An output layer is a layer of an OLS that is output. FIG. 2 depicts three different OLSs, namely OLS 1, OLS 2, and OLS 3. As shown, OLS 1 includes Layer N 231 and Layer N+1 232. Layer N 231 includes pictures 215, 216, 217 and 218, and Layer N+1 232 includes pictures 211, 212, 213, and 214. OLS 2 includes Layer N 231, Layer N+1 232, Layer N+2 233, and Layer N+3 234. Layer N+2 233 includes pictures 241, 242, 243, and 244, and Layer N+3 234 includes pictures 251, 252, 253, and 254. OLS 3 includes Layer N 231, Layer N+1 232, and Layer N+2 233. Despite three OLSs being shown, a different number of OLSs may be used in practical applications. In the illustrated embodiment, none of the OLSs include Layer N+4 235, which contains pictures 261, 262, 263, and 264.

**[0056]** Each of the different OLSs may contain any number of layers. The different OLSs are generated in an effort to accommodate the coding capabilities of a variety of different devices having varying coding capabilities. For example, OLS 1, which contains only two layers, may be generated to accommodate a mobile phone with relatively limited coding capabilities. On the other hand, OLS 2, which contains four layers, may be generated to accommodate a big screen television, which is able to decode higher layers than the mobile phone. OLS 3, which contains three layers, may be generated to accommodate a personal computer, laptop computer, or a tablet

computer, which may be able to decode higher layers than the mobile phone but cannot decode the highest layers like the big screen television.

**[0057]** The layers in FIG. 2 can be all independent from each other. That is, each layer can be coded without using inter-layer prediction (ILP). In this case, the layers are referred to as simulcast layers. One or more of the layers in FIG. 2 may also be coded using ILP. Whether the layers are simulcast layers or whether some of the layers are coded using ILP may be signaled by a flag in a video parameter set (VPS). When some layers use ILP, the layer dependency relationship among layers is also signaled in the VPS.

**[0058]** In an embodiment, when the layers are simulcast layers, only one layer is selected for decoding and output. In an embodiment, when some layers use ILP, all of the layers (e.g., the entire bitstream) are specified to be decoded, and certain layers among the layers are specified to be output layers. The output layer or layers may be, for example, 1) only the highest layer, 2) all the layers, or 3) the highest layer plus a set of indicated lower layers. For example, when the highest layer plus a set of indicated lower layers are designated for output by a flag in the VPS, Layer N+3 234 (which is the highest layer) and Layers N 231 and N+1 232 (which are lower layers) from OLS 2 are output.

**[0059]** Some layers in FIG. 2 may be referred to as primary layers, while other layers may be referred to as auxiliary layers. For example, Layer N 231 and Layer N+1 232 may be referred to as primary layers, and Layer N+2 233 and Layer N+3 234 may be referred to as auxiliary layers. The auxiliary layers may be referred to as an alpha auxiliary layer or a depth auxiliary layer. A primary layer may be associated with an auxiliary layer when auxiliary information is present in the bitstream.

**[0060]** Unfortunately, existing standards have drawbacks. 1. Currently, the syntax element `sdi_view_id_len` is coded as `u(4)`, and the value is required to be in the range of 0 to 15, inclusive. This value specifies the length in bits of the `sdi_view_id_val[ i ]` syntax element, specifying the view ID of the *i*-th layer in the bitstream. However, the length of `sdi_view_id_val[ i ]` shall not be equal to 0, while this is currently allowed.

**[0061]** 2. When some auxiliary information is present in the bitstream, e.g., as indicated by the SDI SEI message (a.k.a., the scalability dimension SEI message), and the depth representation information SEI message or the alpha channel information SEI message, it is unknown which non-auxiliary or primary layers the auxiliary information applies to.

**[0062]** 3. It does not make sense to have a multiview acquisition information SEI message, depth representation information SEI message, or alpha channel information SEI message

present in the bitstream if the scalability dimension information SEI message is not present in the bitstream.

[0063] 4. The multiview acquisition information SEI message contains information for all views present in the bitstream. Therefore, it's meaningless for it to be scalable-nested while this is currently allowed.

[0064] Disclosed herein are techniques that solve one or more of the foregoing problems. For example, the present disclosure provides techniques that use a scalability dimension information (SDI) view identifier length minus L syntax element to prevent a length of an SDI view ID value syntax element, which specifies a view identifier of an i-th layer in a bitstream, from being zero. The disclosed aspects/embodiments further provide techniques that prevent a bitstream from having a multiview acquisition information supplemental enhancement information (SEI) message or an auxiliary information SEI message when an SDI message is not present in the bitstream. The disclosed aspects/embodiments also provide techniques that prevent the multiview acquisition information SEI message from being scalable-nested.

[0065] FIG. 3 illustrates an embodiment of a video bitstream 300. As used herein the video bitstream 300 may also be referred to as a coded video bitstream, a bitstream, or variations thereof. As shown in FIG. 3, the bitstream 300 comprises one or more of the following: decoding capability information (DCI) 302, a video parameter set (VPS) 304, a sequence parameter set (SPS) 306, a picture parameter set (PPS) 308, a picture header (PH) 312, a picture 314, and an SEI message 322. Each of the DCI 302, the VPS 304, the SPS 306, and the PPS 308 may be generically referred to as a parameter set. In an embodiment, other parameter sets not shown in FIG. 3 may also be included in the bitstream 300 such as, for example, an adaption parameter set (APS), which is a syntax structure containing syntax elements that apply to zero or more slices as determined by zero or more syntax elements found in slice headers.

[0066] The DCI 302, which may also be referred to a decoding parameter set (DPS) or decoder parameter set, is a syntax structure containing syntax elements that apply to the entire bitstream. The DCI 302 includes parameters that stay constant for the lifetime of the video bitstream (e.g., bitstream 300), which can translate to the lifetime of a session. The DCI 302 can include profile, level, and sub-profile information to determine a maximum complexity interop point that is guaranteed to be never exceeded, even if splicing of video sequences occurs within a session. It further optionally includes constraint flags, which indicate that the video bitstream will be constraint of the use of certain features as indicated by the values of those flags. With this, a bitstream can be labelled as not using certain tools, which allows among other things for resource allocation in a decoder implementation. Like all parameter sets, the DCI 302 is present

when first referenced, and referenced by the very first picture in a video sequence, implying that it has to be sent among the first network abstraction layer (NAL) units in the bitstream. While multiple DCIs 302 can be in the bitstream, the value of the syntax elements therein cannot be inconsistent when being referenced.

**[0067]** The VPS 304 includes decoding dependency or information for reference picture set construction of enhancement layers. The VPS 304 provides an overall perspective or view of a scalable sequence, including what types of operation points are provided, the profile, tier, and level of the operation points, and some other high-level properties of the bitstream that can be used as the basis for session negotiation and content selection, etc.

**[0068]** In an embodiment, when it is indicated that some of the layers use ILP, the VPS 304 indicates that a total number of OLSs specified by the VPS is equal to the number of layers, indicates that the *i*-th OLS includes the layers with layer indices from 0 to *i*, inclusive, and indicates that for each OLS only the highest layer in the OLS is output.

**[0069]** The SPS 306 contains data that is common to all the pictures in a sequence of pictures (SOP). The SPS 306 is a syntax structure containing syntax elements that apply to zero or more entire CLVSs as determined by the content of a syntax element found in the PPS referred to by a syntax element found in each picture header. In contrast, the PPS 308 contains data that is common to the entire picture. The PPS 308 is a syntax structure containing syntax elements that apply to zero or more entire coded pictures as determined by a syntax element found in each picture header (e.g., PH 312).

**[0070]** The DCI 302, the VPS 304, the SPS 306, and the PPS 308 are contained in different types of Network Abstraction Layer (NAL) units. A NAL unit is a syntax structure containing an indication of the type of data to follow (e.g., coded video data). NAL units are classified into video coding layer (VCL) and non-VCL NAL units. The VCL NAL units contain the data that represents the values of the samples in the video pictures, and the non-VCL NAL units contain any associated additional information such as parameter sets (important data that can apply to a number of VCL NAL units) and supplemental enhancement information (timing information and other supplemental data that may enhance usability of the decoded video signal but are not necessary for decoding the values of the samples in the video pictures).

**[0071]** In an embodiment, the DCI 302 is contained in a non-VCL NAL unit designated as a DCI NAL unit or a DPS NAL unit. That is, the DCI NAL unit has a DCI NAL unit type (NUT) and the DPS NAL unit has a DPS NUT. In an embodiment, the VPS 304 is contained in a non-VCL NAL unit designated as a VPS NAL unit. Therefore, the VPS NAL unit has a VPS NUT. In an embodiment, the SPS 306 is a non-VCL NAL unit designated as a SPS NAL unit.

Therefore, the SPS NAL unit has an SPS NUT. In an embodiment, the PPS 308 is contained in a non-VCL NAL unit designated as a PPS NAL unit. Therefore, the PPS NAL unit has a PPS NUT.

[0072] The PH 312 is a syntax structure containing syntax elements that apply to all slices (e.g., slices 318) of a coded picture (e.g., picture 314). In an embodiment, the PH 312 is in a type of non-VCL NAL unit designated a PH NAL unit. Therefore, the PH NAL unit has a PH NUT (e.g., PH\_NUT).

[0073] In an embodiment, the PH NAL unit associated with the PH 312 has a temporal ID and a layer ID. The temporal ID identifier indicates the position of the PH NAL unit, in time, relative to the other PH NAL units in the bitstream (e.g., bitstream 300). The layer ID indicates the layer (e.g., layer 131 or layer 132) that contains the PH NAL unit. In an embodiment, the temporal ID is similar to, but different from, the picture order count (POC). The POC uniquely identifies each picture in order. In a single layer bitstream, temporal ID and POC would be the same. In a multi-layer bitstream (e.g., see FIG. 1), pictures in the same AU would have different POCs, but the same temporal ID.

[0074] In an embodiment, the PH NAL unit precedes the VCL NAL unit containing the first slice 318 of the associated picture 314. This establishes the association between the PH 312 and the slices 318 of the picture 314 associated with the PH 312 without the need to have a picture header ID signaled in the PH 312 and referred to from the slice header 320. Consequently, it can be inferred that all VCL NAL units between two PHs 312 belong to the same picture 314 and that the picture 314 is associated with the first PH 312 between the two PHs 312. In an embodiment, the first VCL NAL unit that follows a PH 312 contains the first slice 318 of the picture 314 associated with the PH 312.

[0075] In an embodiment, the PH NAL unit follows picture level parameter sets (e.g., the PPS) or higher level parameter sets such as the DCI (a.k.a., the DPS), the VPS, the SPS, the PPS, etc., having both a temporal ID and a layer ID less than the temporal ID and layer ID of the PH NAL unit, respectively. Consequently, those parameter sets are not repeated within a picture or an access unit. Because of this ordering, the PH 312 can be resolved immediately. That is, parameter sets that contain parameters relevant to an entire picture are positioned in the bitstream before the PH NAL unit. Anything that contains parameters for part of a picture is positioned after the PH NAL unit.

[0076] In one alternative, the PH NAL unit follows picture level parameter sets and prefix supplemental enhancement information (SEI) messages, or higher level parameter sets such as the DCI (a.k.a., the DPS), the VPS, the SPS, the PPS, the APS, the SEI message, etc.

[0077] The picture 314 is an array of luma samples in monochrome format or an array of luma samples and two corresponding arrays of chroma samples in 4:2:0, 4:2:2, and 4:4:4 color format.

[0078] The picture 314 may be either a frame or a field. However, in one CVS 316, either all pictures 314 are frames or all pictures 314 are fields. The CVS 316 is a coded video sequence for every coded layer video sequence (CLVS) in the video bitstream 300. Notably, the CVS 316 and the CLVS are the same when the video bitstream 300 includes a single layer. The CVS 316 and the CLVS are only different when the video bitstream 300 includes multiple layers (e.g., as shown in FIGS. 1 and 2).

[0079] Each picture 314 contains one or more slices 318. A slice 318 is an integer number of complete tiles or an integer number of consecutive complete coding tree unit (CTU) rows within a tile of a picture (e.g., picture 314). Each slice 318 is exclusively contained in a single NAL unit (e.g., a VCL NAL unit). A tile (not shown) is a rectangular region of CTUs within a particular tile column and a particular tile row in a picture (e.g., picture 314). A CTU (not shown) is a coding tree block (CTB) of luma samples, two corresponding CTBs of chroma samples of a picture that has three sample arrays, or a CTB of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to code the samples. A CTB (not shown) is an  $N \times N$  block of samples for some value of  $N$  such that the division of a component into CTBs is a partitioning. A block (not shown) is an  $M \times N$  ( $M$ -column by  $N$ -row) array of samples (e.g., pixels), or an  $M \times N$  array of transform coefficients.

[0080] In an embodiment, each slice 318 contains a slice header 320. A slice header 320 is the part of the coded slice 318 containing the data elements pertaining to all tiles or CTU rows within a tile represented in the slice 318. That is, the slice header 320 contains information about the slice 318 such as, for example, the slice type, which of the reference pictures will be used, and so on.

[0081] The pictures 314 and their slices 318 comprise data associated with the images or video being encoded or decoded. Thus, the pictures 314 and their slices 318 may be simply referred to as the payload or data being carried in the bitstream 300.

[0082] The bitstream 300 also contains one or more SEI messages, such as SEI message 322, SEI message 326, and SEI message 328. The SEI messages contain supplemental enhancement information. SEI messages can contain various types of data that indicate the timing of the video pictures or describe various properties of the coded video or how the coded video can be used or enhanced. SEI messages are also defined that can contain arbitrary user-defined data. SEI messages do not affect the core decoding process, but can indicate how the video is recommended

to be post-processed or displayed. Some other high-level properties of the video content are conveyed in video usability information (VUI), such as the indication of the color space for interpretation of the video content. As new color spaces have been developed, such as for high dynamic range and wide color gamut video, additional VUI identifiers have been added to indicate them.

**[0083]** In an embodiment, the SEI message 322 may be an SDI SEI message. The SDI SEI message may be used to indicate which primary layers are associated with an auxiliary layer when auxiliary information is present in a bitstream. For example, the SDI SEI message may include one or more syntax elements 324 to indicate which primary layers are associated with the auxiliary layer when the auxiliary information is present in the bitstream. A discussion of various SEI messages and the syntax elements included in those SEI messages is provided below.

**[0084]** In an embodiment, the SEI message 326 is a multiview information SEI message, such as a multiview acquisition information SEI message. When present in the bitstream 300, the multiview information SEI message includes one or more syntax elements 324 that specify various parameters of the acquisition environment, e.g., intrinsic and extrinsic camera parameters. These parameters are useful for view warping and interpolation.

**[0085]** In an embodiment, the SEI message 328 may be an auxiliary information SEI message, such as a depth representation information SEI message or an alpha channel information SEI message. When present in the bitstream 300, the depth representation information SEI message includes one or more syntax elements 324 that specify various depth representation for depth views for the purpose of processing decoded texture and depth view components prior to rendering on a three dimensional (3D) display, such as view synthesis. The SEI message may be associated with an instantaneous decoder refresh (IDR) access unit for the purpose of random access. When present in the bitstream 300, the alpha channel information SEI message includes one or more syntax elements 324 that provide information about alpha channel sample values and post-processing applied to the decoded alpha plane auxiliary pictures, and one or more associated primary pictures. Blending is the process of combining two images into a single image. An image to be blended is associated with an auxiliary image identified as an alpha plane. The alpha channel information SEI message may be used to specify how the pixel values of the image to be blended are converted to another image comprising interpretation values.

**[0086]** Those skilled in the art will appreciate that the bitstream 300 may contain other parameters and information in practical applications.

[0087] To solve the above problems, methods as summarized below are disclosed. The techniques should be considered as examples to explain the general concepts and should not be interpreted in a narrow way. Furthermore, these techniques can be applied individually or combined in any manner.

[0088] **Example 1**

[0089] 1) To solve problem 1, in one example, instead of signaling the length of view ID syntax elements, e.g., via the syntax element `sdi_view_id_len`, the value of the length minus L (e.g.,  $L=1$ ) is signaled, e.g., via the syntax element `sdi_view_id_len_minusL`.

[0090] a. In one example, furthermore, the syntax element may be coded as an unsigned integer using N bits.

[0091] i. In one example, N may be equal to 4.

[0092] ii. Alternatively, the syntax may be coded as a fixed-pattern bit string using N bits, or signed integer using N bits, or truncated binary, or a signed integer K-th (e.g.,  $K=0$ ) order Exp-Golomb-coded syntax element, or an unsigned integer M-th (e.g.,  $M=0$ ) order Exp-Golomb-coded syntax element.

[0093] b. In one example, alternatively, still signal the length, e.g., via the syntax element `sdi_view_id_len`, but it is constrained that the value of syntax element shall not be equal to 0.

[0094] **Example 2**

[0095] 2) To solve problem 2, it is proposed that an auxiliary layer (i.e., a layer having the corresponding `sdi_aux_id[ i ]` equal to 1 or 2) may be applied to one or more associated layers.

[0096] a. In one example, one or more syntax elements indicating the associated layers for each auxiliary layer may be signaled in the scalability dimension information SEI message.

[0097] i. In one example, the associated layers are specified by layer IDs.

[0098] ii. In another example, the associated layers are specified by layer indices.

[0099] iii. In another example, the indication whether the auxiliary layer is applied to one or more associated layers may be specified by one or more syntax elements for the associated layers.

[00100] 1. In one example, a syntax element may be used to indicate whether the auxiliary layer is applied to all the associated layers.

[00101] 2. In one example, a syntax element may be used to indicate whether the auxiliary layer is applied to a specific associated layer.

[00102] a. In one example, one or more primary layers are indicated by the syntax elements.

[00103] i. In one example, all the primary layers may be indicated by the syntax elements.

[00104] ii. In one example, only the primary layers of which the layer index is smaller than the layer index of the auxiliary layer may be indicated by the syntax elements.

[00105] iii. In one example, only the primary layers of which the layer index is larger than the layer index of the auxiliary layer may be indicated by the syntax elements.

[00106] b. In one example, the syntax element is coded as a flag.

[00107] b. Alternatively, it is proposed that the associated one or more layers for each auxiliary layer may be derived without being explicitly signaled.

[00108] i. In one example, the associated layers for each auxiliary layer may be the layer having `nuh_layer_id` equal to the `nuh_layer_id` of the auxiliary layer plus  $N_1, N_2, \dots$ , and  $N_k$ , respectively, where  $k$  is an integer and  $N_i \neq N_j$  for any  $i, j$  ( $i \neq j$ ) in the range of 1 to  $k$ , inclusive.

[00109] 1. In one example,  $k$  is equal to 1 and  $N_1$  may be equal to 1, or 2, or -1, or -2.

[00110] 2. In one example,  $k$  is greater than 1.

[00111] a. In one example,  $k$  is equal to 2 and  $N_1 = 1, N_2 = 2$ .

[00112] ii. In one example, the associated layers for each auxiliary layer may be the layer having layer index equal to the layer index of the auxiliary layer plus  $N_1, N_2, \dots$ , and  $N_k$ , respectively, where  $k$  is an integer and  $N_i \neq N_j$  for any  $i, j$  ( $i \neq j$ ) in the range of 1 to  $k$ , inclusive.

[00113] 1. In one example,  $k$  is equal to 1 and  $N_1$  may be equal to 1, or 2, or -1, or -2.

[00114] 2. In one example,  $k$  is greater than 1.

[00115] a. In one example,  $k$  is equal to 2 and  $N_1 = 1, N_2 = 2$ .

[00116] c. Alternatively, indications of the associated layers of each auxiliary layer may be explicitly signaled as one or a group of syntax elements in the scalability dimension information SEI message.

[00117] d. Alternatively, indications of the associated layers of an auxiliary information SEI message (e.g., depth representation information or alpha channel information) may be explicitly signaled by one or more syntax elements in the auxiliary information SEI message.

[00118] i. In one example, the auxiliary information SEI message may refer to the depth representation information SEI message or the alpha channel information SEI message.

[00119] ii. In one example, the one or more syntax elements may indicate layer ID values of the associated layers.

[00120] 1. In one example, the layer IDs indicated by the syntax elements may be required to be less than or equal to the maximum layer ID value, i.e., `vps_layer_id[ vps_max_layers_minus1 ]` or `vps_layer_id[ sdi_max_layers_minus1 ]`.

[00121] iii. In one example, the one or more syntax elements may indicate layer index values of the associated layers.

[00122] 1. In one example, the layer indices indicated by the syntax elements may be required to be less than the maximum number of layers in the bitstream (e.g.,  $sdi\_max\_layers\_minus1$  plus 1 or  $vps\_max\_layers\_minus1$  plus 1).

[00123] iv. In one example, indication of whether one or multiple layers are associated with auxiliary layers may be signaled.

[00124] 1. In one example, one syntax element may be used to specify whether an auxiliary information SEI message applies to all layers.

[00125] a. In one example,  $auxiliary\_all\_layer\_flag$  equal to X (X being 1 or 0) may specify that the auxiliary information SEI message is applied to all associated primary layers.

[00126] 2. In one example, one or more syntax elements may be used to specify whether the auxiliary information SEI message is applied to one or more layers.

[00127] a. In one example, N syntax element may be used to specify whether the auxiliary information SEI message is applied to N layers, wherein each syntax element is used for each layer.

[00128] i. In one example, the syntax element may be coded as a flag using 1 bit.

[00129] b. In one example, one syntax element may be used to specify whether the auxiliary information SEI message is applied to one or more layers.

[00130] i. In one example, the syntax element may be K-th (e.g., K=0) Exp-Golomb coded.

[00131] ii. In one example, the syntax element equal to 5 specifies that the auxiliary information SEI message is applied to 0-th and 2nd layer but not applied to 1st layer.

[00132] 1. Alternatively, denote N as the number of the layers. the syntax element equal to 5 specifies that the auxiliary information SEI message is applied to (N – 1)-th and (N – 3)-nd layer but not applied to (N – 2)-th layer.

[00133] c. The above syntax elements may be conditionally signaled, e.g., only when the auxiliary information SEI message is not applied to all layers,

[00134] e. In one example, indication of number of associated layers of auxiliary pictures for one layer may be signaled in the bitstream.

[00135] f. In one example, the above syntax elements may be signaled using unsigned integer using N bits, or, fixed-pattern bit string using N bits, or signed integer using N bits, or truncated binary, or signed integer K-th (e.g., K=0) order Exp-Golomb-coded syntax element, or unsigned integer M-th (e.g., M=0) order Exp-Golomb-coded syntax element.

[00136] g. In one example, indications of number of associated layers of auxiliary pictures and/or associated layers of auxiliary pictures may be conditionally signaled, e.g., only when the i-th layer in  $bitstreamInScope$  contains auxiliary pictures (e.g.,  $sdi\_aux\_id[i] > 0$ ). The

bitstreamInScope (a.k.a., bitstream in scope) is defined as a sequence of AUs that consists, in decoding order, of an initial AU containing an SDI SEI message followed by zero or more subsequent AUs up to, but not including, any subsequent AU that contains another SDI SEI message.

**[00137] Example 3**

**[00138]** 3) To solve problem 3, a requirement of bitstream conformance is added that multiview or auxiliary information SEI message shall not be present in a CVS that does not have a scalability dimension information SEI message.

**[00139]** a. Furthermore, the multiview information SEI message may refer to the multiview acquisition information SEI message.

**[00140]** b. Furthermore, the auxiliary information SEI message may refer to the depth representation information SEI message or the alpha channel information SEI message.

**[00141]** c. Alternatively, a requirement of bitstream conformance is added that when the multiview or auxiliary information SEI message is present in the bitstream, at least one of sdi\_multiview\_info\_flag and sdi\_auxiliary\_info\_flag of the scalability dimension information SEI message is required to be equal to 1.

**[00142] Example 4**

**[00143]** 4) To solve problem 4, in one example, a requirement of bitstream conformance is added that the multiview acquisition information SEI message shall not be scalable-nested.

**[00144]** a. Alternatively, it is specified that an SEI message that has payloadType equal to 179 (multiview acquisition) shall not be contained in a scalable nesting SEI message.

**[00145]** Below are some example embodiments for some of the examples summarized above. Each embodiment can be applied to VVC. Most relevant parts that have been added or modified are depicted in a bold italic font, and some of the deleted parts are depicted in an italic font. There may be some other changes that are editorial in nature and thus not highlighted.

**[00146]** Each scalability dimension SEI message syntax described below includes one or more syntax elements. A syntax element may be, for example, one or more values, flags, variables, phrases, indications, indices, mappings, data elements, or a combination thereof included in the scalability dimension SEI message syntax disclosed herein. In an embodiment, the syntax elements may be organized into a group of values, flags, variables, phrases, indications, indices, mappings, and/or data elements.

**[00147] Embodiment 1**

**[00148]** Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                     | Descriptor |
|------------------------------------------------------------|------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)       |
| <b>sdi_multiview_info_flag</b>                             | u(1)       |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)       |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_len_minus1</b>                              | u(4)       |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_val[ i ]</b>                                | u(v)       |
| if( sdi_auxiliary_info_flag )                              |            |
| <b>sdi_aux_id[ i ]</b>                                     | u(8)       |
| }                                                          |            |
| }                                                          |            |
| }                                                          |            |

**[00149]** Scalability dimension SEI message semantics

**[00150]** The scalability dimension SEI message provides the scalability dimension information for each layer in bitstreamInScope (defined below), such as 1) when bitstreamInScope may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in bitstreamInScope, the auxiliary ID of each layer.

**[00151]** The bitstreamInScope is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

**[00152]** **sdi\_max\_layers\_minus1** plus 1 indicates the maximum number of layers in bitstreamInScope.

**[00153]** **sdi\_multiview\_info\_flag** equal to 1 indicates that bitstreamInScope may be a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_multiview\_info\_flag** equal to 0 indicates that bitstreamInScope is not a

multiview bitstream and the `sdi_view_id_val[ ]` syntax elements are not present in the scalability dimension SEI message.

**[00154]** `sdi_auxiliary_info_flag` equal to 1 indicates that there may be auxiliary information carried by one or more layers in `bitstreamInScope` and the `sdi_aux_id[ ]` syntax elements are present in the scalability dimension SEI message. `sdi_auxiliary_info_flag` equal to 0 indicates that there is no auxiliary information carried by one or more layers in `bitstreamInScope` and the `sdi_aux_id[ ]` syntax elements are not present in the scalability dimension SEI message.

**[00155]** `sdi_view_id_len_minus1 plus 1` specifies the length, in bits, of the `sdi_view_id_val[ i ]` syntax element.

**[00156]** `sdi_view_id_val[ i ]` specifies the view ID of the *i*-th layer in `bitstreamInScope`. The length of the `sdi_view_id_val[ i ]` syntax element is `sdi_view_id_len_minus1 + 1` bits. When not present, the value of `sdi_view_id_val[ i ]` is inferred to be equal to 0.

**[00157]** `sdi_aux_id[ i ]` equal to 0 indicates that the *i*-th layer in `bitstreamInScope` does not contain auxiliary pictures. `sdi_aux_id[ i ]` greater than 0 indicates the type of auxiliary pictures in the *i*-th layer in `bitstreamInScope` as specified in Table 1.

Table 1 – Mapping of `sdi_aux_id[ i ]` to the type of auxiliary pictures

| <code>sdi_aux_id[ i ]</code> | Name      | Type of auxiliary pictures |
|------------------------------|-----------|----------------------------|
| 1                            | AUX_ALPHA | Alpha plane                |
| 2                            | AUX_DEPTH | Depth picture              |
| 3..127                       |           | Reserved                   |
| 128..159                     |           | Unspecified                |
| 160..255                     |           | Reserved                   |

**[00158]** NOTE 1 – The interpretation of auxiliary pictures associated with `sdi_aux_id` in the range of 128 to 159, inclusive, is specified through means other than the `sdi_aux_id` value.

**[00159]** `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of `sdi_aux_id[ i ]` in the range of 0 to 255, inclusive.

**[00160]** **Embodiment 2**

**[00161]** Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                     | Descriptor |
|------------------------------------------------------------|------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)       |
| <b>sdi_multiview_info_flag</b>                             | u(1)       |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)       |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_len_minus1</b>                              | u(4)       |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_val[ i ]</b>                                | u(v)       |
| if( sdi_auxiliary_info_flag )                              |            |
| sdi_aux_id[ i ]                                            | u(8)       |
| }                                                          |            |
| }                                                          |            |
| if( sdi_auxiliary_info_flag ) {                            |            |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |            |
| sdi_aux_id[ i ]                                            |            |
| }                                                          |            |
| }                                                          |            |
| }                                                          |            |

**[00162]** Scalability dimension SEI message semantics

**[00163]** The scalability dimension SEI message provides the scalability dimension information for each layer in bitstreamInScope (defined below), such as 1) when bitstreamInScope may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in bitstreamInScope, the auxiliary ID of each layer.

**[00164]** The bitstreamInScope is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs,

including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

[00165] **sdi\_max\_layers\_minus1 plus 1** indicates the maximum number of layers in bitstreamInScope.

[00166] **sdi\_multiview\_info\_flag** equal to 1 indicates that bitstreamInScope may be a multiview bitstream and the sdi\_view\_id\_val[ ] syntax elements are present in the scalability dimension SEI message. sdi\_multiview\_flag equal to 0 indicates that bitstreamInScope is not a multiview bitstream and the sdi\_view\_id\_val[ ] syntax elements are not present in the scalability dimension SEI message.

[00167] **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in bitstreamInScope and the sdi\_aux\_id[ ] syntax elements are present in the scalability dimension SEI message. sdi\_auxiliary\_info\_flag equal to 0 indicates that there is no auxiliary information carried by one or more layers in bitstreamInScope and the sdi\_aux\_id[ ] syntax elements are not present in the scalability dimension SEI message.

[00168] **sdi\_view\_id\_len\_minus1 plus 1** specifies the length, in bits, of the sdi\_view\_id\_val[ i ] syntax element.

[00169] **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in bitstreamInScope. The length of the sdi\_view\_id\_val[ i ] syntax element is sdi\_view\_id\_len\_minus1 + 1 bits. When not present, the value of sdi\_view\_id\_val[ i ] is inferred to be equal to 0.

[00170] **sdi\_aux\_id[ i ]** equal to 0 indicates that the i-th layer in bitstreamInScope does not contain auxiliary pictures. sdi\_aux\_id[ i ] greater than 0 indicates the type of auxiliary pictures in the i-th layer in bitstreamInScope as specified in Table 1.

Table 1 – Mapping of sdi\_aux\_id[ i ] to the type of auxiliary pictures

| sdi_aux_id[ i ] | Name      | Type of auxiliary pictures |
|-----------------|-----------|----------------------------|
| 1               | AUX_ALPHA | Alpha plane                |
| 2               | AUX_DEPTH | Depth picture              |
| 3..127          |           | Reserved                   |
| 128..159        |           | Unspecified                |
| 160..255        |           | Reserved                   |

[00171] NOTE 1 – The interpretation of auxiliary pictures associated with sdi\_aux\_id in the range of 128 to 159, inclusive, is specified through means other than the sdi\_aux\_id value.

[00172] `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of `sdi_aux_id[ i ]` in the range of 0 to 255, inclusive.

[00173] **Embodiment 3**

[00174] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                     | Descriptor |
|------------------------------------------------------------|------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)       |
| <b>sdi_multiview_info_flag</b>                             | u(1)       |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)       |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_len</b>                                     | u(4)       |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_val[ i ]</b>                                | u(v)       |
| if( sdi_auxiliary_info_flag )                              |            |
| <b>sdi_aux_id[ i ]</b>                                     | u(8)       |
| }                                                          |            |
| }                                                          |            |
| }                                                          |            |

[00175] Scalability dimension SEI message semantics

[00176] The scalability dimension SEI message provides the scalability dimension information for each layer in `bitstreamInScope` (defined below), such as 1) when `bitstreamInScope` may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in `bitstreamInScope`, the auxiliary ID of each layer.

[00177] The `bitstreamInScope` is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs,

including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

**[00178]** **sdi\_max\_layers\_minus1** plus 1 indicates the maximum number of layers in bitstreamInScope.

**[00179]** **sdi\_multiview\_info\_flag** equal to 1 indicates that bitstreamInScope may be a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_multiview\_flag** equal to 0 indicates that bitstreamInScope is not a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are not present in the scalability dimension SEI message.

**[00180]** **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in bitstreamInScope and the **sdi\_aux\_id[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_auxiliary\_info\_flag** equal to 0 indicates that there is no auxiliary information carried by one or more layers in bitstreamInScope and the **sdi\_aux\_id[ ]** syntax elements are not present in the scalability dimension SEI message.

**[00181]** **sdi\_view\_id\_len** specifies the length, in bits, of the **sdi\_view\_id\_val[ i ]** syntax element. *When present, sdi\_view\_id\_len shall not be equal to 0.*

**[00182]** **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in bitstreamInScope. The length of the **sdi\_view\_id\_val[ i ]** syntax element is **sdi\_view\_id\_len** bits. When not present, the value of **sdi\_view\_id\_val[ i ]** is inferred to be equal to 0.

**[00183]** **sdi\_aux\_id[ i ]** equal to 0 indicates that the i-th layer in bitstreamInScope does not contain auxiliary pictures. **sdi\_aux\_id[ i ]** greater than 0 indicates the type of auxiliary pictures in the i-th layer in bitstreamInScope as specified in Table 1.

Table 1 – Mapping of **sdi\_aux\_id[ i ]** to the type of auxiliary pictures

| <b>sdi_aux_id[ i ]</b> | <b>Name</b> | <b>Type of auxiliary pictures</b> |
|------------------------|-------------|-----------------------------------|
| 1                      | AUX_ALPHA   | Alpha plane                       |
| 2                      | AUX_DEPTH   | Depth picture                     |
| 3..127                 |             | Reserved                          |
| 128..159               |             | Unspecified                       |
| 160..255               |             | Reserved                          |

**[00184]** NOTE 1 – The interpretation of auxiliary pictures associated with **sdi\_aux\_id** in the range of 128 to 159, inclusive, is specified through means other than the **sdi\_aux\_id** value.

[00185] `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of `sdi_aux_id[ i ]` in the range of 0 to 255, inclusive.

[00186] **Embodiment 4**

[00187] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                     | Descriptor  |
|------------------------------------------------------------|-------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)        |
| <b>sdi_multiview_info_flag</b>                             | u(1)        |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)        |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |             |
| if( sdi_multiview_info_flag )                              |             |
| <b>sdi_view_id_len</b>                                     | u(4)        |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |             |
| if( sdi_multiview_info_flag )                              |             |
| <b>sdi_view_id_val[ i ]</b>                                | u(v)        |
| if( sdi_auxiliary_info_flag ) {                            |             |
| <b>sdi_aux_id[ i ]</b>                                     | u(8)        |
| <i>if( sdi_aux_id[ i ] &gt; 0 ) {</i>                      |             |
| <i>sdi_associated_primary_id[ i ]</i>                      | <i>u(6)</i> |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |

[00188] Scalability dimension SEI message semantics

[00189] The scalability dimension SEI message provides the scalability dimension information for each layer in `bitstreamInScope` (defined below), such as 1) when `bitstreamInScope` may be a multiview bitstream, the view ID of each layer; and 2) when there

may be auxiliary information (such as depth or alpha) carried by one or more layers in bitstreamInScope, the auxiliary ID of each layer.

[00190] The bitstreamInScope is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

[00191] **sdi\_max\_layers\_minus1** plus 1 indicates the maximum number of layers in bitstreamInScope.

[00192] **sdi\_multiview\_info\_flag** equal to 1 indicates that bitstreamInScope may be a multiview bitstream and the sdi\_view\_id\_val[ ] syntax elements are present in the scalability dimension SEI message. sdi\_multiview\_flag equal to 0 indicates that bitstreamInScope is not a multiview bitstream and the sdi\_view\_id\_val[ ] syntax elements are not present in the scalability dimension SEI message.

[00193] **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in bitstreamInScope and the sdi\_aux\_id[ ] syntax elements are present in the scalability dimension SEI message. sdi\_auxiliary\_info\_flag equal to 0 indicates that there is no auxiliary information carried by one or more layers in bitstreamInScope and the sdi\_aux\_id[ ] syntax elements are not present in the scalability dimension SEI message.

[00194] **sdi\_view\_id\_len** specifies the length, in bits, of the sdi\_view\_id\_val[ i ] syntax element.

[00195] *Alternatively, the following applies:*

[00196] *sdi\_view\_id\_len specifies the length, in bits, of the sdi\_view\_id\_val[ i ] syntax element. When present, sdi\_view\_id\_len shall not be equal to 0.*

[00197] **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in bitstreamInScope. The length of the sdi\_view\_id\_val[ i ] syntax element is sdi\_view\_id\_len bits. When not present, the value of sdi\_view\_id\_val[ i ] is inferred to be equal to 0.

[00198] **sdi\_aux\_id[ i ]** equal to 0 indicates that the i-th layer in bitstreamInScope does not contain auxiliary pictures. sdi\_aux\_id[ i ] greater than 0 indicates the type of auxiliary pictures in the i-th layer in bitstreamInScope as specified in Table 1.

Table 1 – Mapping of sdi\_aux\_id[ i ] to the type of auxiliary pictures

| sdi_aux_id[ i ] | Name      | Type of auxiliary pictures |
|-----------------|-----------|----------------------------|
| 1               | AUX_ALPHA | Alpha plane                |
| 2               | AUX_DEPTH | Depth picture              |
| 3..127          |           | Reserved                   |

|          |  |             |
|----------|--|-------------|
| 128..159 |  | Unspecified |
| 160..255 |  | Reserved    |

[00199] NOTE 1 – The interpretation of auxiliary pictures associated with `sdi_aux_id` in the range of 128 to 159, inclusive, is specified through means other than the `sdi_aux_id` value.

[00200] `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of `sdi_aux_id[ i ]` in the range of 0 to 255, inclusive.

[00201] *`sdi_associated_primary_id[ i ]` specifies the layer ID of the associated primary layer of the *i*-th layer, which is an auxiliary layer.*

[00202] Embodiment 5

[00203] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                     | Descriptor  |
|------------------------------------------------------------|-------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)        |
| <b>sdi_multiview_info_flag</b>                             | u(1)        |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)        |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |             |
| if( sdi_multiview_info_flag )                              |             |
| <b>sdi_view_id_len</b>                                     | u(4)        |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |             |
| if( sdi_multiview_info_flag )                              |             |
| <b>sdi_view_id_val[ i ]</b>                                | u(v)        |
| if( sdi_auxiliary_info_flag ) {                            |             |
| <b>sdi_aux_id[ i ]</b>                                     | u(8)        |
| <i>if( sdi_aux_id[ i ] &gt; 0 ) {</i>                      |             |
| <i>sdi_associated_primary_idx[ i ]</i>                     | <i>u(6)</i> |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |

|   |  |
|---|--|
| } |  |
|---|--|

**[00204]** Scalability dimension SEI message semantics

**[00205]** The scalability dimension SEI message provides the scalability dimension information for each layer in `bitstreamInScope` (defined below), such as 1) when `bitstreamInScope` may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in `bitstreamInScope`, the auxiliary ID of each layer.

**[00206]** The `bitstreamInScope` is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

**[00207]** `sdi_max_layers_minus1` plus 1 indicates the maximum number of layers in `bitstreamInScope`.

**[00208]** `sdi_multiview_info_flag` equal to 1 indicates that `bitstreamInScope` may be a multiview bitstream and the `sdi_view_id_val[ ]` syntax elements are present in the scalability dimension SEI message. `sdi_multiview_flag` equal to 0 indicates that `bitstreamInScope` is not a multiview bitstream and the `sdi_view_id_val[ ]` syntax elements are not present in the scalability dimension SEI message.

**[00209]** `sdi_auxiliary_info_flag` equal to 1 indicates that there may be auxiliary information carried by one or more layers in `bitstreamInScope` and the `sdi_aux_id[ ]` syntax elements are present in the scalability dimension SEI message. `sdi_auxiliary_info_flag` equal to 0 indicates that there is no auxiliary information carried by one or more layers in `bitstreamInScope` and the `sdi_aux_id[ ]` syntax elements are not present in the scalability dimension SEI message.

**[00210]** `sdi_view_id_len` specifies the length, in bits, of the `sdi_view_id_val[ i ]` syntax element.

**[00211]** `sdi_view_id_val[ i ]` specifies the view ID of the i-th layer in `bitstreamInScope`. The length of the `sdi_view_id_val[ i ]` syntax element is `sdi_view_id_len` bits. When not present, the value of `sdi_view_id_val[ i ]` is inferred to be equal to 0.

**[00212]** `sdi_aux_id[ i ]` equal to 0 indicates that the i-th layer in `bitstreamInScope` does not contain auxiliary pictures. `sdi_aux_id[ i ]` greater than 0 indicates the type of auxiliary pictures in the i-th layer in `bitstreamInScope` as specified in Table 1.

Table 1 – Mapping of `sdi_aux_id[ i ]` to the type of auxiliary pictures

| <b>sdi_aux_id[ i ]</b> | <b>Name</b> | <b>Type of auxiliary pictures</b> |
|------------------------|-------------|-----------------------------------|
| 1                      | AUX_ALPHA   | Alpha plane                       |
| 2                      | AUX_DEPTH   | Depth picture                     |
| 3..127                 |             | Reserved                          |
| 128..159               |             | Unspecified                       |
| 160..255               |             | Reserved                          |

**[00213]** NOTE 1 – The interpretation of auxiliary pictures associated with sdi\_aux\_id in the range of 128 to 159, inclusive, is specified through means other than the sdi\_aux\_id value.

**[00214]** sdi\_aux\_id[ i ] shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of sdi\_aux\_id[ i ] shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of sdi\_aux\_id[ i ] in the range of 0 to 255, inclusive.

**[00215]** *sdi\_associated\_primary\_idx[ i ] specifies the layer index of the associated primary layer of the i-th layer, which is an auxiliary layer. The value of sdi\_aux\_id[ sdi\_associated\_primary\_layer\_idx[ i ] ] shall be equal to 0.*

**[00216]** NOTE 2 – *An auxiliary layer describes a property of and applies to its associated primary layers.*

**[00217]** Embodiment 6

**[00218]** Scalability dimension SEI message syntax

| <b>scalability_dimension( payloadSize ) {</b>              | <b>Descriptor</b> |
|------------------------------------------------------------|-------------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)              |
| <b>sdi_multiview_info_flag</b>                             | u(1)              |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)              |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |                   |
| if( sdi_multiview_info_flag )                              |                   |
| <b>sdi_view_id_len</b>                                     | u(4)              |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |                   |
| if( sdi_multiview_info_flag )                              |                   |
| <b>sdi_view_id_val[ i ]</b>                                | u(v)              |

|                                                                                            |                    |
|--------------------------------------------------------------------------------------------|--------------------|
| <i>if( sdi_auxiliary_info_flag ) {</i>                                                     |                    |
| <b><i>sdi_aux_id[ i ]</i></b>                                                              | <b><i>u(8)</i></b> |
| <i>if ( sdi_aux_id[ i ] &gt; 0 ) {</i>                                                     |                    |
| <b><i>sdi_num_associated_primary_layers_minus1[ i ]</i></b>                                | <b><i>u(6)</i></b> |
| <i>for (j = 0; j &lt;= sdi_num_associated_primary_layers_minus1[ i ];</i><br><i>j++) {</i> |                    |
| <b><i>sdi_associated_primary_layer_id[ i ][ j ]</i></b>                                    | <b><i>u(6)</i></b> |
| <i>}</i>                                                                                   |                    |
| <i>}</i>                                                                                   |                    |
| <i>}</i>                                                                                   |                    |
| <i>}</i>                                                                                   |                    |
| <i>}</i>                                                                                   |                    |
| <i>}</i>                                                                                   |                    |

**[00219]** Scalability dimension SEI message semantics

**[00220]** The scalability dimension SEI message provides the scalability dimension information for each layer in *bitstreamInScope* (defined below), such as 1) when *bitstreamInScope* may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in *bitstreamInScope*, the auxiliary ID of each layer.

**[00221]** The *bitstreamInScope* is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

**[00222]** ***sdi\_max\_layers\_minus1*** plus 1 indicates the maximum number of layers in *bitstreamInScope*.

**[00223]** ***sdi\_multiview\_info\_flag*** equal to 1 indicates that *bitstreamInScope* may be a multiview bitstream and the *sdi\_view\_id\_val[ ]* syntax elements are present in the scalability dimension SEI message. *sdi\_multiview\_flag* equal to 0 indicates that *bitstreamInScope* is not a multiview bitstream and the *sdi\_view\_id\_val[ ]* syntax elements are not present in the scalability dimension SEI message.

[00224] **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in bitstreamInScope and the **sdi\_aux\_id[ i ]** syntax elements are present in the scalability dimension SEI message. **sdi\_auxiliary\_info\_flag** equal to 0 indicates that there is no auxiliary information carried by one or more layers in bitstreamInScope and the **sdi\_aux\_id[ i ]** syntax elements are not present in the scalability dimension SEI message.

[00225] **sdi\_view\_id\_len** specifies the length, in bits, of the **sdi\_view\_id\_val[ i ]** syntax element.

[00226] **sdi\_view\_id\_val[ i ]** specifies the view ID of the *i*-th layer in bitstreamInScope. The length of the **sdi\_view\_id\_val[ i ]** syntax element is **sdi\_view\_id\_len** bits. When not present, the value of **sdi\_view\_id\_val[ i ]** is inferred to be equal to 0.

[00227] **sdi\_aux\_id[ i ]** equal to 0 indicates that the *i*-th layer in bitstreamInScope does not contain auxiliary pictures. **sdi\_aux\_id[ i ]** greater than 0 indicates the type of auxiliary pictures in the *i*-th layer in bitstreamInScope as specified in Table 1.

Table 1 – Mapping of **sdi\_aux\_id[ i ]** to the type of auxiliary pictures

| <b>sdi_aux_id[ i ]</b> | <b>Name</b> | <b>Type of auxiliary pictures</b> |
|------------------------|-------------|-----------------------------------|
| 1                      | AUX_ALPHA   | Alpha plane                       |
| 2                      | AUX_DEPTH   | Depth picture                     |
| 3..127                 |             | Reserved                          |
| 128..159               |             | Unspecified                       |
| 160..255               |             | Reserved                          |

[00228] NOTE 1 – The interpretation of auxiliary pictures associated with **sdi\_aux\_id** in the range of 128 to 159, inclusive, is specified through means other than the **sdi\_aux\_id** value.

[00229] **sdi\_aux\_id[ i ]** shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of **sdi\_aux\_id[ i ]** shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of **sdi\_aux\_id[ i ]** in the range of 0 to 255, inclusive.

[00230] *If **sdi\_aux\_id[ i ]** is equal to 0, the *i*-th layer is referred to as a primary layer. Otherwise, the *i*-th layer is referred to as an auxiliary layer. When **sdi\_aux\_id[ i ]** is equal to 1, the *i*-th layer is also referred to as an alpha auxiliary layer. When **sdi\_aux\_id[ i ]** is equal to 2, the *i*-th layer is also referred to as a depth auxiliary layer.*

[00231] *sdi\_num\_associated\_primary\_layers\_minus1[i]* plus 1 indicates the number of associated primary layers of the *i*-th layer, which is an auxiliary layer. The value of *sdi\_num\_associated\_primary\_layers\_minus1[i]* shall be less than the total number of primary layers.

[00232] *sdi\_associated\_primary\_layer\_id[i][j]* indicates the layer ID of the associated layer of the *i*-th layer, which is an auxiliary layer.

[00233] Embodiment 7

[00234] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                                                       | Descriptor  |
|----------------------------------------------------------------------------------------------|-------------|
| <b>sdi_max_layers_minus1</b>                                                                 | u(6)        |
| <b>sdi_multiview_info_flag</b>                                                               | u(1)        |
| <b>sdi_auxiliary_info_flag</b>                                                               | u(1)        |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) {                                   |             |
| if( sdi_multiview_info_flag )                                                                |             |
| <b>sdi_view_id_len</b>                                                                       | u(4)        |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {                                              |             |
| if( sdi_multiview_info_flag )                                                                |             |
| <b>sdi_view_id_val[ i ]</b>                                                                  | u(v)        |
| if( sdi_auxiliary_info_flag ) {                                                              |             |
| <b>sdi_aux_id[ i ]</b>                                                                       | u(8)        |
| <i>if ( sdi_aux_id[ i ] &gt; 0 ) {</i>                                                       |             |
| <i>sdi_num_associated_primary_layers_minus1[ i ]</i>                                         | <i>u(6)</i> |
| <i>for ( j = 0; j &lt;= sdi_num_associated_primary_layers_minus1[ i ];</i><br><i>j++ ) {</i> |             |
| <i>sdi_associated_primary_layer_idx[ i ][ j ]</i>                                            | <i>u(6)</i> |
| <i>}</i>                                                                                     |             |
| <i>}</i>                                                                                     |             |
| <i>}</i>                                                                                     |             |
| <i>}</i>                                                                                     |             |
| <i>}</i>                                                                                     |             |

|   |  |
|---|--|
| } |  |
|---|--|

**[00235]** Scalability dimension SEI message semantics

**[00236]** The scalability dimension SEI message provides the scalability dimension information for each layer in `bitstreamInScope` (defined below), such as 1) when `bitstreamInScope` may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in `bitstreamInScope`, the auxiliary ID of each layer.

**[00237]** The `bitstreamInScope` is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

**[00238]** `sdi_max_layers_minus1` plus 1 indicates the maximum number of layers in `bitstreamInScope`.

**[00239]** `sdi_multiview_info_flag` equal to 1 indicates that `bitstreamInScope` may be a multiview bitstream and the `sdi_view_id_val[ ]` syntax elements are present in the scalability dimension SEI message. `sdi_multiview_flag` equal to 0 indicates that `bitstreamInScope` is not a multiview bitstream and the `sdi_view_id_val[ ]` syntax elements are not present in the scalability dimension SEI message.

**[00240]** `sdi_auxiliary_info_flag` equal to 1 indicates that there may be auxiliary information carried by one or more layers in `bitstreamInScope` and the `sdi_aux_id[ ]` syntax elements are present in the scalability dimension SEI message. `sdi_auxiliary_info_flag` equal to 0 indicates that there is no auxiliary information carried by one or more layers in `bitstreamInScope` and the `sdi_aux_id[ ]` syntax elements are not present in the scalability dimension SEI message.

**[00241]** `sdi_view_id_len` specifies the length, in bits, of the `sdi_view_id_val[ i ]` syntax element.

**[00242]** `sdi_view_id_val[ i ]` specifies the view ID of the i-th layer in `bitstreamInScope`. The length of the `sdi_view_id_val[ i ]` syntax element is `sdi_view_id_len` bits. When not present, the value of `sdi_view_id_val[ i ]` is inferred to be equal to 0.

**[00243]** `sdi_aux_id[ i ]` equal to 0 indicates that the i-th layer in `bitstreamInScope` does not contain auxiliary pictures. `sdi_aux_id[ i ]` greater than 0 indicates the type of auxiliary pictures in the i-th layer in `bitstreamInScope` as specified in Table 1.

Table 2 – Mapping of `sdi_aux_id[ i ]` to the type of auxiliary pictures

| sdi_aux_id[ i ] | Name      | Type of auxiliary pictures |
|-----------------|-----------|----------------------------|
| 1               | AUX_ALPHA | Alpha plane                |
| 2               | AUX_DEPTH | Depth picture              |
| 3..127          |           | Reserved                   |
| 128..159        |           | Unspecified                |
| 160..255        |           | Reserved                   |

[00244] NOTE 1 – The interpretation of auxiliary pictures associated with sdi\_aux\_id in the range of 128 to 159, inclusive, is specified through means other than the sdi\_aux\_id value.

[00245] sdi\_aux\_id[ i ] shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of sdi\_aux\_id[ i ] shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of sdi\_aux\_id[ i ] in the range of 0 to 255, inclusive.

[00246] *If sdi\_aux\_id[ i ] is equal to 0, the i-th layer is referred to as a primary layer. Otherwise, the i-th layer is referred to as an auxiliary layer. When sdi\_aux\_id[ i ] is equal to 1, the i-th layer is also referred to as an alpha auxiliary layer. When sdi\_aux\_id[ i ] is equal to 2, the i-th layer is also referred to as a depth auxiliary layer.*

[00247] sdi\_num\_associated\_primary\_layers\_minus1[ i ] plus 1 indicates the number of associated primary layers of the i-th layer, which is an auxiliary layer. The value of sdi\_num\_associated\_primary\_layers\_minus1[ i ] shall be less than the total number of primary layers.

[00248] sdi\_associated\_primary\_layer\_idx[ i ][ j ] specifies the layer index of the j-th associated primary layer of the i-th layer, which is an auxiliary layer. The value of sdi\_aux\_id[ sdi\_associated\_primary\_layer\_idx[ i ][ j ] ] shall be equal to 0.

[00249] NOTE 2 – An auxiliary layer describes a property of and applies to its associated primary layers.

[00250] Embodiment 8

[00251] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) { | Descriptor |
|----------------------------------------|------------|
| sdi_max_layers_minus1                  | u(6)       |
| sdi_multiview_info_flag                | u(1)       |

|                                                            |             |
|------------------------------------------------------------|-------------|
| <b>sdi_auxiliary_info_flag</b>                             | u(1)        |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |             |
| if( sdi_multiview_info_flag )                              |             |
| <b>sdi_view_id_len</b>                                     | u(4)        |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |             |
| if( sdi_multiview_info_flag )                              |             |
| <b>sdi_view_id_val[ i ]</b>                                | u(v)        |
| if( sdi_auxiliary_info_flag ) {                            |             |
| <b>sdi_aux_id[ i ]</b>                                     | u(8)        |
| <i>if ( sdi_aux_id[ i ] &gt; 0 ) {</i>                     |             |
| <i>for ( j = 0; j &lt;= i - 1; j++ ) {</i>                 |             |
| <i>if ( sdi_aux_id[ j ] == 0 ) {</i>                       |             |
| <b>sdi_associated_primary_layer_flag[ I ][ j ]</b>         | <b>u(6)</b> |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |
| <i>}</i>                                                   |             |

**[00252]** Scalability dimension SEI message semantics

**[00253]** The scalability dimension SEI message provides the scalability dimension information for each layer in bitstreamInScope (defined below), such as 1) when bitstreamInScope may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in bitstreamInScope, the auxiliary ID of each layer.

**[00254]** The bitstreamInScope is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs,

including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

[00255] **sdi\_max\_layers\_minus1** plus 1 indicates the maximum number of layers in bitstreamInScope.

[00256] **sdi\_multiview\_info\_flag** equal to 1 indicates that bitstreamInScope may be a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_multiview\_flag** equal to 0 indicates that bitstreamInScope is not a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are not present in the scalability dimension SEI message.

[00257] **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in bitstreamInScope and the **sdi\_aux\_id[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_auxiliary\_info\_flag** equal to 0 indicates that there is no auxiliary information carried by one or more layers in bitstreamInScope and the **sdi\_aux\_id[ ]** syntax elements are not present in the scalability dimension SEI message.

[00258] **sdi\_view\_id\_len** specifies the length, in bits, of the **sdi\_view\_id\_val[ i ]** syntax element.

[00259] **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in bitstreamInScope. The length of the **sdi\_view\_id\_val[ i ]** syntax element is **sdi\_view\_id\_len** bits. When not present, the value of **sdi\_view\_id\_val[ i ]** is inferred to be equal to 0.

[00260] **sdi\_aux\_id[ i ]** equal to 0 indicates that the i-th layer in bitstreamInScope does not contain auxiliary pictures. **sdi\_aux\_id[ i ]** greater than 0 indicates the type of auxiliary pictures in the i-th layer in bitstreamInScope as specified in Table 1.

Table 1 – Mapping of **sdi\_aux\_id[ i ]** to the type of auxiliary pictures

| <b>sdi_aux_id[ i ]</b> | <b>Name</b> | <b>Type of auxiliary pictures</b> |
|------------------------|-------------|-----------------------------------|
| 1                      | AUX_ALPHA   | Alpha plane                       |
| 2                      | AUX_DEPTH   | Depth picture                     |
| 3..127                 |             | Reserved                          |
| 128..159               |             | Unspecified                       |
| 160..255               |             | Reserved                          |

[00261] NOTE 1 – The interpretation of auxiliary pictures associated with **sdi\_aux\_id** in the range of 128 to 159, inclusive, is specified through means other than the **sdi\_aux\_id** value.

[00262] `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of `sdi_aux_id[ i ]` in the range of 0 to 255, inclusive.

[00263] *If `sdi_aux_id[ i ]` is equal to 0, the *i*-th layer is referred to as a primary layer. Otherwise, the *i*-th layer is referred to as an auxiliary layer. When `sdi_aux_id[ i ]` is equal to 1, the *i*-th layer is also referred to as an alpha auxiliary layer. When `sdi_aux_id[ i ]` is equal to 2, the *i*-th layer is also referred to as a depth auxiliary layer.*

[00264] *`sdi_associated_primary_layer_flag[ i ][ j ]` equal to 1 specifies the *j*-th layer is the associated primary layer of the *i*-th layer, which is an auxiliary layer.*

[00265] *NOTE 2 – An auxiliary layer describes a property of and applies to its associated primary layers.*

[00266] **Embodiment 9**

[00267] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                                                                         | Descriptor  |
|----------------------------------------------------------------------------------------------------------------|-------------|
| <b>sdi_max_layers_minus1</b>                                                                                   | u(6)        |
| <b>sdi_multiview_info_flag</b>                                                                                 | u(1)        |
| <b>sdi_auxiliary_info_flag</b>                                                                                 | u(1)        |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) {                                                     |             |
| if( sdi_multiview_info_flag )                                                                                  |             |
| <b>sdi_view_id_len</b>                                                                                         | u(4)        |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {                                                                |             |
| if( sdi_multiview_info_flag )                                                                                  |             |
| <b>sdi_view_id_val[ i ]</b>                                                                                    | u(v)        |
| if( sdi_auxiliary_info_flag ) {                                                                                |             |
| <b>sdi_aux_id[ i ]</b>                                                                                         | u(8)        |
| <i>if ( <code>sdi_aux_id[ i ]</code> &gt; 0 ) {</i>                                                            |             |
| <i>for ( <code>j</code> = 0; <code>j</code> &lt;= <code>sdi_max_layers_minus1</code>; <code>j</code>++ ) {</i> |             |
| <i>if ( <code>sdi_aux_id[ j ]</code> == 0 ) {</i>                                                              |             |
| <b><i>sdi_associated_primary_layer_flag[ i ][ j ]</i></b>                                                      | <b>u(6)</b> |
| <i>}</i>                                                                                                       |             |
| <i>}</i>                                                                                                       |             |

|   |  |
|---|--|
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |

**[00268]** Scalability dimension SEI message semantics

**[00269]** The scalability dimension SEI message provides the scalability dimension information for each layer in `bitstreamInScope` (defined below), such as 1) when `bitstreamInScope` may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in `bitstreamInScope`, the auxiliary ID of each layer.

**[00270]** The `bitstreamInScope` is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

**[00271]** `sdi_max_layers_minus1` plus 1 indicates the maximum number of layers in `bitstreamInScope`.

**[00272]** `sdi_multiview_info_flag` equal to 1 indicates that `bitstreamInScope` may be a multiview bitstream and the `sdi_view_id_val[ ]` syntax elements are present in the scalability dimension SEI message. `sdi_multiview_flag` equal to 0 indicates that `bitstreamInScope` is not a multiview bitstream and the `sdi_view_id_val[ ]` syntax elements are not present in the scalability dimension SEI message.

**[00273]** `sdi_auxiliary_info_flag` equal to 1 indicates that there may be auxiliary information carried by one or more layers in `bitstreamInScope` and the `sdi_aux_id[ ]` syntax elements are present in the scalability dimension SEI message. `sdi_auxiliary_info_flag` equal to 0 indicates that there is no auxiliary information carried by one or more layers in `bitstreamInScope` and the `sdi_aux_id[ ]` syntax elements are not present in the scalability dimension SEI message.

**[00274]** `sdi_view_id_len` specifies the length, in bits, of the `sdi_view_id_val[ i ]` syntax element.

[00275] **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in bitstreamInScope. The length of the sdi\_view\_id\_val[ i ] syntax element is sdi\_view\_id\_len bits. When not present, the value of sdi\_view\_id\_val[ i ] is inferred to be equal to 0.

[00276] **sdi\_aux\_id[ i ]** equal to 0 indicates that the i-th layer in bitstreamInScope does not contain auxiliary pictures. sdi\_aux\_id[ i ] greater than 0 indicates the type of auxiliary pictures in the i-th layer in bitstreamInScope as specified in Table 1.

Table 1 – Mapping of sdi\_aux\_id[ i ] to the type of auxiliary pictures

| sdi_aux_id[ i ] | Name      | Type of auxiliary pictures |
|-----------------|-----------|----------------------------|
| 1               | AUX_ALPHA | Alpha plane                |
| 2               | AUX_DEPTH | Depth picture              |
| 3..127          |           | Reserved                   |
| 128..159        |           | Unspecified                |
| 160..255        |           | Reserved                   |

[00277] NOTE 1 – The interpretation of auxiliary pictures associated with sdi\_aux\_id in the range of 128 to 159, inclusive, is specified through means other than the sdi\_aux\_id value.

[00278] sdi\_aux\_id[ i ] shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of sdi\_aux\_id[ i ] shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of sdi\_aux\_id[ i ] in the range of 0 to 255, inclusive.

[00279] *If sdi\_aux\_id[ i ] is equal to 0, the i-th layer is referred to as a primary layer. Otherwise, the i-th layer is referred to as an auxiliary layer. When sdi\_aux\_id[ i ] is equal to 1, the i-th layer is also referred to as an alpha auxiliary layer. When sdi\_aux\_id[ i ] is equal to 2, the i-th layer is also referred to as a depth auxiliary layer.*

[00280] *sdi\_associated\_primary\_layer\_flag[ i ][ j ] equal to 1 specifies the j-th layer is the associated primary layer of the i-th layer, which is an auxiliary layer.*

[00281] NOTE 2 – *An auxiliary layer describes a property of and applies to its associated primary layers.*

[00282] **Embodiment 10**

[00283] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) { | Descriptor |
|----------------------------------------|------------|
| <b>sdi_max_layers_minus1</b>           | u(6)       |
| <b>sdi_multiview_info_flag</b>         | u(1)       |
| <b>sdi_auxiliary_info_flag</b>         | u(1)       |

|                                                             |             |
|-------------------------------------------------------------|-------------|
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) {  |             |
| if( sdi_multiview_info_flag )                               |             |
| <b>sdi_view_id_len</b>                                      | u(4)        |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {             |             |
| if( sdi_multiview_info_flag )                               |             |
| <b>sdi_view_id_val[ i ]</b>                                 | u(v)        |
| if( sdi_auxiliary_info_flag ) {                             |             |
| <b>sdi_aux_id[ i ]</b>                                      | u(8)        |
| <i>if ( sdi_aux_id[ i ] &gt; 0 ) {</i>                      |             |
| <i>sdi_all_associated_primary_layers_flag[ i ]</i>          | <i>u(1)</i> |
| <i>if ( sdi_all_associated_primary_layers_flag == 0 ) {</i> |             |
| <i>for ( j = 0; j &lt;= sdi_max_layers_minus1; j++ ) {</i>  |             |
| <i>if ( sdi_aux_id[ j ] == 0 ) {</i>                        |             |
| <i>sdi_associated_primary_layer_flag[ i ][ j ]</i>          | <i>u(6)</i> |
| <i>}</i>                                                    |             |
| <i>}</i>                                                    |             |
| <i>}</i>                                                    |             |
| <i>}</i>                                                    |             |
| <i>}</i>                                                    |             |
| <i>}</i>                                                    |             |
| <i>}</i>                                                    |             |
| <i>}</i>                                                    |             |

[00284] Scalability dimension SEI message semantics

[00285] The scalability dimension SEI message provides the scalability dimension information for each layer in bitstreamInScope (defined below), such as 1) when bitstreamInScope may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in bitstreamInScope, the auxiliary ID of each layer.

[00286] The **bitstreamInScope** is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

[00287] **sdi\_max\_layers\_minus1** plus 1 indicates the maximum number of layers in **bitstreamInScope**.

[00288] **sdi\_multiview\_info\_flag** equal to 1 indicates that **bitstreamInScope** may be a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_multiview\_flag** equal to 0 indicates that **bitstreamInScope** is not a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are not present in the scalability dimension SEI message.

[00289] **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in **bitstreamInScope** and the **sdi\_aux\_id[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_auxiliary\_info\_flag** equal to 0 indicates that there is no auxiliary information carried by one or more layers in **bitstreamInScope** and the **sdi\_aux\_id[ ]** syntax elements are not present in the scalability dimension SEI message.

[00290] **sdi\_view\_id\_len** specifies the length, in bits, of the **sdi\_view\_id\_val[ i ]** syntax element.

[00291] **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in **bitstreamInScope**. The length of the **sdi\_view\_id\_val[ i ]** syntax element is **sdi\_view\_id\_len** bits. When not present, the value of **sdi\_view\_id\_val[ i ]** is inferred to be equal to 0.

[00292] **sdi\_aux\_id[ i ]** equal to 0 indicates that the i-th layer in **bitstreamInScope** does not contain auxiliary pictures. **sdi\_aux\_id[ i ]** greater than 0 indicates the type of auxiliary pictures in the i-th layer in **bitstreamInScope** as specified in Table 1.

Table 1 – Mapping of **sdi\_aux\_id[ i ]** to the type of auxiliary pictures

| <b>sdi_aux_id[ i ]</b> | <b>Name</b> | <b>Type of auxiliary pictures</b> |
|------------------------|-------------|-----------------------------------|
| 1                      | AUX_ALPHA   | Alpha plane                       |
| 2                      | AUX_DEPTH   | Depth picture                     |
| 3..127                 |             | Reserved                          |
| 128..159               |             | Unspecified                       |
| 160..255               |             | Reserved                          |

[00293] NOTE 1 – The interpretation of auxiliary pictures associated with `sdi_aux_id` in the range of 128 to 159, inclusive, is specified through means other than the `sdi_aux_id` value.

[00294] `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of `sdi_aux_id[ i ]` shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of `sdi_aux_id[ i ]` in the range of 0 to 255, inclusive.

[00295] *If `sdi_aux_id[ i ]` is equal to 0, the *i*-th layer is referred to as a primary layer. Otherwise, the *i*-th layer is referred to as an auxiliary layer. When `sdi_aux_id[ i ]` is equal to 1, the *i*-th layer is also referred to as an alpha auxiliary layer. When `sdi_aux_id[ i ]` is equal to 2, the *i*-th layer is also referred to as a depth auxiliary layer.*

[00296] *`sdi_all_associated_primary_layers_flag[ i ]` equal to 1 specifies the *i*-th layer, which is an auxiliary layer, is applied to all the primary layers. `sdi_all_associated_primary_layers_flag[ i ]` equal to 0 specifies the *i*-th layer, which is an auxiliary layer, is applied to one or more primary layers.*

[00297] *`sdi_associated_primary_layer_flag[ i ][ j ]` equal to 1 specifies the *j*-th layer is the associated primary layer of the *i*-th layer, which is an auxiliary layer. `sdi_associated_primary_layer_flag[ i ][ j ]` equal to 0 specifies the *i*-th layer, which is an auxiliary layer, is not applied to the *j*-th primary layer. When `sdi_all_associated_primary_layers_flag[ i ]` equal to 1, `sdi_associated_primary_layer_flag[ i ][ j ]` is inferred to be 1.*

[00298] NOTE 1 – An auxiliary layer describes a property of and applies to its associated primary layers.

[00299] Embodiment 11

[00300] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                     | Descriptor |
|------------------------------------------------------------|------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)       |
| <b>sdi_multiview_info_flag</b>                             | u(1)       |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)       |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_len</b>                                     | u(4)       |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |            |

|                               |      |
|-------------------------------|------|
| if( sdi_multiview_info_flag ) |      |
| <b>sdi_view_id_val[ i ]</b>   | u(v) |
| if( sdi_auxiliary_info_flag ) |      |
| <b>sdi_aux_id[ i ]</b>        | u(8) |
| }                             |      |
| }                             |      |
| }                             |      |

**[00301]** Scalability dimension SEI message semantics

**[00302]** The scalability dimension SEI message provides the scalability dimension information for each layer in bitstreamInScope (defined below), such as 1) when bitstreamInScope may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in bitstreamInScope, the auxiliary ID of each layer.

**[00303]** The bitstreamInScope is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

**[00304]** **sdi\_max\_layers\_minus1** plus 1 indicates the maximum number of layers in bitstreamInScope.

**[00305]** **sdi\_multiview\_info\_flag** equal to 1 indicates that bitstreamInScope may be a multiview bitstream and the sdi\_view\_id\_val[ ] syntax elements are present in the scalability dimension SEI message. sdi\_multiview\_flag equal to 0 indicates that bitstreamInScope is not a multiview bitstream and the sdi\_view\_id\_val[ ] syntax elements are not present in the scalability dimension SEI message.

**[00306]** **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in bitstreamInScope and the sdi\_aux\_id[ ] syntax elements are present in the scalability dimension SEI message. sdi\_auxiliary\_info\_flag equal to 0 indicates that there is no auxiliary information carried by one or more layers in bitstreamInScope and the sdi\_aux\_id[ ] syntax elements are not present in the scalability dimension SEI message.

**[00307]** **sdi\_view\_id\_len** specifies the length, in bits, of the sdi\_view\_id\_val[ i ] syntax element.

[00308] **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in bitstreamInScope. The length of the sdi\_view\_id\_val[ i ] syntax element is sdi\_view\_id\_len bits. When not present, the value of sdi\_view\_id\_val[ i ] is inferred to be equal to 0.

[00309] **sdi\_aux\_id[ i ]** equal to 0 indicates that the i-th layer in bitstreamInScope does not contain auxiliary pictures. sdi\_aux\_id[ i ] greater than 0 indicates the type of auxiliary pictures in the i-th layer in bitstreamInScope as specified in Table 1. *The associated layer for each auxiliary layer (i.e., sdi\_aux\_id[ i ] equal to 1 or 2) is the layer having nuh\_layer\_id equal to the nuh\_layer\_id of the auxiliary layer plus 1.*

Table 1 – Mapping of sdi\_aux\_id[ i ] to the type of auxiliary pictures

| <b>sdi_aux_id[ i ]</b> | <b>Name</b> | <b>Type of auxiliary pictures</b> |
|------------------------|-------------|-----------------------------------|
| 1                      | AUX_ALPHA   | Alpha plane                       |
| 2                      | AUX_DEPTH   | Depth picture                     |
| 3..127                 |             | Reserved                          |
| 128..159               |             | Unspecified                       |
| 160..255               |             | Reserved                          |

[00310] NOTE 1 – The interpretation of auxiliary pictures associated with sdi\_aux\_id in the range of 128 to 159, inclusive, is specified through means other than the sdi\_aux\_id value.

[00311] **sdi\_aux\_id[ i ]** shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of sdi\_aux\_id[ i ] shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of sdi\_aux\_id[ i ] in the range of 0 to 255, inclusive.

[00312] **Embodiment 12**

[00313] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                     | <b>Descript<br/>or</b> |
|------------------------------------------------------------|------------------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)                   |
| <b>sdi_multiview_info_flag</b>                             | u(1)                   |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)                   |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |                        |
| if( sdi_multiview_info_flag )                              |                        |
| <b>sdi_view_id_len</b>                                     | u(4)                   |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |                        |
| if( sdi_multiview_info_flag )                              |                        |

|                               |      |
|-------------------------------|------|
| <b>sdi_view_id_val[ i ]</b>   | u(v) |
| if( sdi_auxiliary_info_flag ) |      |
| <b>sdi_aux_id[ i ]</b>        | u(8) |
| }                             |      |
| }                             |      |
| }                             |      |

**[00314]** Scalability dimension SEI message semantics

**[00315]** The scalability dimension SEI message provides the scalability dimension information for each layer in bitstreamInScope (defined below), such as 1) when bitstreamInScope may be a multiview bitstream, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers in bitstreamInScope, the auxiliary ID of each layer.

**[00316]** The bitstreamInScope is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

**[00317]** **sdi\_max\_layers\_minus1** plus 1 indicates the maximum number of layers in bitstreamInScope.

**[00318]** **sdi\_multiview\_info\_flag** equal to 1 indicates that bitstreamInScope may be a multiview bitstream and the sdi\_view\_id\_val[ ] syntax elements are present in the scalability dimension SEI message. sdi\_multiview\_flag equal to 0 indicates that bitstreamInScope is not a multiview bitstream and the sdi\_view\_id\_val[ ] syntax elements are not present in the scalability dimension SEI message.

**[00319]** **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in bitstreamInScope and the sdi\_aux\_id[ ] syntax elements are present in the scalability dimension SEI message. sdi\_auxiliary\_info\_flag equal to 0 indicates that there is no auxiliary information carried by one or more layers in bitstreamInScope and the sdi\_aux\_id[ ] syntax elements are not present in the scalability dimension SEI message.

**[00320]** **sdi\_view\_id\_len** specifies the length, in bits, of the sdi\_view\_id\_val[ i ] syntax element.

**[00321]** **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in bitstreamInScope. The length of the sdi\_view\_id\_val[ i ] syntax element is sdi\_view\_id\_len bits. When not present, the value of sdi\_view\_id\_val[ i ] is inferred to be equal to 0.

[00322] **sdi\_aux\_id[ i ]** equal to 0 indicates that the *i*-th layer in bitstreamInScope does not contain auxiliary pictures. **sdi\_aux\_id[ i ]** greater than 0 indicates the type of auxiliary pictures in the *i*-th layer in bitstreamInScope as specified in Table 1.

[00323] *When sdi\_aux\_id[ i ] is equal to 1 or 2, let nuhLayerId be the value of the nuh\_layer\_id of the i-th layer, the following applies:*

- *The i-th layer is referred to as an alpha auxiliary layer or a depth auxiliary layer, respectively.*
- *The value of sdi\_aux\_id[ GeneralLayerIdx[ nuhLayerId + 1 ] ] shall be equal to 0, and the layer with nuh\_layer\_id equal to nuhLayerId + 1 is referred to as the primary layer associated with the i-th layer.*

**NOTE 2** – *An alpha auxiliary layer or a depth auxiliary layer applies to its associated primary layer that is not an auxiliary layer.*

Table 1 – Mapping of **sdi\_aux\_id[ i ]** to the type of auxiliary pictures

| <b>sdi_aux_id[ i ]</b> | <b>Name</b> | <b>Type of auxiliary pictures</b> |
|------------------------|-------------|-----------------------------------|
| 1                      | AUX_ALPHA   | Alpha plane                       |
| 2                      | AUX_DEPTH   | Depth picture                     |
| 3..127                 |             | Reserved                          |
| 128..159               |             | Unspecified                       |
| 160..255               |             | Reserved                          |

[00324] **NOTE 1** – The interpretation of auxiliary pictures associated with **sdi\_aux\_id** in the range of 128 to 159, inclusive, is specified through means other than the **sdi\_aux\_id** value.

[00325] **sdi\_aux\_id[ i ]** shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of **sdi\_aux\_id[ i ]** shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of **sdi\_aux\_id[ i ]** in the range of 0 to 255, inclusive.

[00326] **Embodiment 13**

[00327] Depth representation information SEI message

[00328] Depth representation information SEI message syntax

|                                                                           |                    |
|---------------------------------------------------------------------------|--------------------|
| depth_representation_info( payloadSize ) {                                | <b>Descriptor</b>  |
| <i>depth_representation_primary_layer_id</i>                              | <b><i>u(6)</i></b> |
| <b>z_near_flag</b>                                                        | u(1)               |
| <b>z_far_flag</b>                                                         | u(1)               |
| <b>d_min_flag</b>                                                         | u(1)               |
| <b>d_max_flag</b>                                                         | u(1)               |
| <b>depth_representation_type</b>                                          | ue(v)              |
| if( d_min_flag    d_max_flag )                                            |                    |
| <b>disparity_ref_view_id</b>                                              | ue(v)              |
| if( z_near_flag )                                                         |                    |
| depth_rep_info_element( ZNearSign, ZNearExp, ZNearMantissa, ZNearManLen ) |                    |
| if( z_far_flag )                                                          |                    |
| depth_rep_info_element( ZFarSign, ZFarExp, ZFarMantissa, ZFarManLen )     |                    |
| if( d_min_flag )                                                          |                    |
| depth_rep_info_element( DMinSign, DMinExp, DMinMantissa, DMinManLen )     |                    |
| if( d_max_flag )                                                          |                    |
| depth_rep_info_element( DMaxSign, DMaxExp, DMaxMantissa, DMaxManLen )     |                    |
| if( depth_representation_type == 3 ) {                                    |                    |
| <b>depth_nonlinear_representation_num_minus1</b>                          | ue(v)              |
| for( i = 1; i <= depth_nonlinear_representation_num_minus1 + 1; i++ )     |                    |
| <b>depth_nonlinear_representation_model[ i ]</b>                          |                    |
| }                                                                         |                    |
| }                                                                         |                    |

| depth_rep_info_element( OutSign, OutExp, OutMantissa, OutManLen ) { | Descriptor |
|---------------------------------------------------------------------|------------|
| e                                                                   | u(1)       |
| da_exponent                                                         | u(7)       |
| da_mantissa_len_minus1                                              | u(5)       |
| da_mantissa                                                         | u(v)       |
| }                                                                   |            |

[00329] Depth representation information SEI message semantics

[00330] The syntax elements in the depth representation information SEI message specify various parameters for auxiliary pictures of type AUX\_DEPTH for the purpose of processing decoded primary and auxiliary pictures prior to rendering on a 3D display, such as view synthesis. Specifically, depth or disparity ranges for depth pictures are specified.

[00331] When present, the depth representation information SEI message shall be associated with one or more layers with sdi\_aux\_id value equal to AUX\_DEPTH. The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the depth representation information SEI message applies.

[00332] When present, the depth representation information SEI message may be included in any access unit. It is recommended that, when present, the SEI message is included for the purpose of random access in an access unit in which the coded picture with nuh\_layer\_id equal to targetLayerId is an Intra Random Access Picture (IRAP) picture.

[00333] For an auxiliary picture with sdi\_aux\_id[ targetLayerId ] equal to AUX\_DEPTH, an associated primary picture, if any, is a picture in the same access unit having sdi\_aux\_id[ nuhLayerIdB ] equal to 0 such that ScalabilityId[ LayerIdxInVps[ targetLayerId ] ][ j ] is equal to ScalabilityId[ LayerIdxInVps[ nuhLayerIdB ] ][ j ] for all values of j in the range of 0 to 2, inclusive, and 4 to 15, inclusive.

[00334] The information indicated in the SEI message applies to all the pictures with nuh\_layer\_id equal to targetLayerId from the access unit containing the SEI message up to but excluding the next picture, in decoding order, associated with a depth representation information SEI message applicable to targetLayerId or to the end of the CLVS of the nuh\_layer\_id equal to targetLayerId, whichever is earlier in decoding order.

[00335] *depth\_representation\_primary\_layer\_id specifies the nuh\_layer\_id value of the associated primary layer to which the depth representation information SEI message applies.*

[00336] **z\_near\_flag** equal to 0 specifies that the syntax elements specifying the nearest depth value are not present in the syntax structure. **z\_near\_flag** equal to 1 specifies that the syntax elements specifying the nearest depth value are present in the syntax structure.

[00337] **z\_far\_flag** equal to 0 specifies that the syntax elements specifying the farthest depth value are not present in the syntax structure. **z\_far\_flag** equal to 1 specifies that the syntax elements specifying the farthest depth value are present in the syntax structure.

[00338] **d\_min\_flag** equal to 0 specifies that the syntax elements specifying the minimum disparity value are not present in the syntax structure. **d\_min\_flag** equal to 1 specifies that the syntax elements specifying the minimum disparity value are present in the syntax structure.

[00339] **d\_max\_flag** equal to 0 specifies that the syntax elements specifying the maximum disparity value are not present in the syntax structure. **d\_max\_flag** equal to 1 specifies that the syntax elements specifying the maximum disparity value are present in the syntax structure.

[00340] **depth\_representation\_type** specifies the representation definition of decoded luma samples of auxiliary pictures as specified in Table Y1. In Table Y1, disparity specifies the horizontal displacement between two texture views and Z value specifies the distance from a camera.

[00341] The variable maxVal is set equal to  $(1 \ll (8 + \text{sps\_bitdepth\_minus8})) - 1$ , where **sps\_bitdepth\_minus8** is the value included in or inferred for the active SPS of the layer with **nuh\_layer\_id** equal to **targetLayerId**.

Table Y1 – Definition of **depth\_representation\_type**

| <b>depth_representation_type</b> | <b>Interpretation</b>                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                                | Each decoded luma sample value of an auxiliary picture represents an inverse of Z value that is uniformly quantized into the range of 0 to maxVal, inclusive.<br><br>When <b>z_far_flag</b> is equal to 1, the luma sample value equal to 0 represents the inverse of ZFar (specified below). When <b>z_near_flag</b> is equal to 1, the luma sample value equal to maxVal represents the inverse of ZNear (specified below). |
| 1                                | Each decoded luma sample value of an auxiliary picture represents disparity that is uniformly quantized into the range of 0 to maxVal, inclusive.                                                                                                                                                                                                                                                                             |

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | When d_min_flag is equal to 1, the luma sample value equal to 0 represents DMin (specified below). When d_max_flag is equal to 1, the luma sample value equal to maxVal represents DMax (specified below).                                                                                                                                                                                                                                                   |
| 2            | Each decoded luma sample value of an auxiliary picture represents a Z value uniformly quantized into the range of 0 to maxVal, inclusive.<br>When z_far_flag is equal to 1, the luma sample value equal to 0 corresponds to ZFar (specified below). When z_near_flag is equal to 1, the luma sample value equal to maxVal represents ZNear (specified below).                                                                                                |
| 3            | Each decoded luma sample value of an auxiliary picture represents a nonlinearly mapped disparity, normalized in range from 0 to maxVal, as specified by depth_nonlinear_representation_num_minus1 and depth_nonlinear_representation_model[i].<br>When d_min_flag is equal to 1, the luma sample value equal to 0 represents DMin (specified below). When d_max_flag is equal to 1, the luma sample value equal to maxVal represents DMax (specified below). |
| Other values | Reserved for future use                                                                                                                                                                                                                                                                                                                                                                                                                                      |

**[00342]** disparity\_ref\_view\_id specifies the ViewId value against which the disparity values are derived.

**[00343]** NOTE 1 – disparity\_ref\_view\_id is present only if d\_min\_flag is equal to 1 or d\_max\_flag is equal to 1 and is useful for depth\_representation\_type values equal to 1 and 3.

**[00344]** The variables in the x column of Table Y2 are derived from the respective variables in the s, e, n and v columns of Table Y2 as follows:

- If the value of e is in the range of 0 to 127, exclusive, x is set equal to  $(-1)^s * 2^{e-31} * (1 + n \div 2^v)$ .
- Otherwise (e is equal to 0), x is set equal to  $(-1)^s * 2^{-(30+v)} * n$ .

**[00345]** NOTE 1 – The above specification is similar to that found in IEC 60559:1989.

Table Y2 – Association between depth parameter variables and syntax elements

| <b>x</b> | <b>s</b>  | <b>e</b> | <b>n</b>      | <b>v</b>    |
|----------|-----------|----------|---------------|-------------|
| ZNear    | ZNearSign | ZNearExp | ZNearMantissa | ZNearManLen |
| ZFar     | ZFarSign  | ZFarExp  | ZFarMantissa  | ZFarManLen  |
| DMax     | DMaxSign  | DMaxExp  | DMaxMantissa  | DMaxManLen  |
| DMin     | DMinSign  | DMinExp  | DMinMantissa  | DMinManLen  |

**[00346]** The DMin and DMax values, when present, are specified in units of a luma sample width of the coded picture with ViewId equal to ViewId of the auxiliary picture.

**[00347]** The units for the ZNear and ZFar values, when present, are identical but unspecified.

**[00348]** **depth\_nonlinear\_representation\_num\_minus1** plus 2 specifies the number of piece-wise linear segments for mapping of depth values to a scale that is uniformly quantized in terms of disparity.

**[00349]** **depth\_nonlinear\_representation\_model[ i ]** for *i* ranging from 0 to **depth\_nonlinear\_representation\_num\_minus1 + 2**, inclusive, specify the piece-wise linear segments for mapping of decoded luma sample values of an auxiliary picture to a scale that is uniformly quantized in terms of disparity. The values of **depth\_nonlinear\_representation\_model[ 0 ]** and **depth\_nonlinear\_representation\_model[ depth\_nonlinear\_representation\_num\_minus1 + 2 ]** are both inferred to be equal to 0.

**[00350]** NOTE 2 – When **depth\_representation\_type** is equal to 3, an auxiliary picture contains nonlinearly transformed depth samples. The variable **DepthLUT[ i ]**, as specified below, is used to transform decoded depth sample values from the nonlinear representation to the linear representation, i.e., uniformly quantized disparity values. The shape of this transform is defined by means of line-segment approximation in two-dimensional linear-disparity-to-nonlinear-disparity space. The first ( 0, 0 ) and the last ( **maxVal**, **maxVal** ) nodes of the curve are predefined. Positions of additional nodes are transmitted in form of deviations (**depth\_nonlinear\_representation\_model[ i ]**) from the straight-line curve. These deviations are uniformly distributed along the whole range of 0 to **maxVal**, inclusive, with spacing depending on the value of **nonlinear\_depth\_representation\_num\_minus1**.

**[00351]** The variable **DepthLUT[ i ]** for *i* in the range of 0 to **maxVal**, inclusive, is specified as follows:

```

for( k = 0; k <= depth_nonlinear_representation_num_minus1 + 1; k++ ) {
    pos1 = ( maxVal * k ) / (depth_nonlinear_representation_num_minus1 + 2 )
    dev1 = depth_nonlinear_representation_model[ k ]
    pos2 = ( maxVal * ( k + 1 ) ) / (depth_nonlinear_representation_num_minus1 + 2 )
    dev2 = depth_nonlinear_representation_model[ k + 1 ]   (X)

    x1 = pos1 - dev1
    y1 = pos1 + dev1
    x2 = pos2 - dev2
    y2 = pos2 + dev2

    for( x = Max( x1, 0 ); x <= Min( x2, maxVal ); x++ )
        DepthLUT[ x ] = Clip3( 0, maxVal, Round( ( ( x - x1 ) * ( y2 - y1 ) ) ÷ ( x2 -
x1 ) + y1 ) )
}

```

**[00352]** When `depth_representation_type` is equal to 3, `DepthLUT[ dS ]` for all decoded luma sample values `dS` of an auxiliary picture in the range of 0 to `maxVal`, inclusive, represents disparity that is uniformly quantized into the range of 0 to `maxVal`, inclusive.

**[00353]** The syntax structure specifies the value of an element in the depth representation information SEI message.

**[00354]** The syntax structure sets the values of the `OutSign`, `OutExp`, `OutMantissa` and `OutManLen` variables that represent a floating-point value. When the syntax structure is included in another syntax structure, the variable names `OutSign`, `OutExp`, `OutMantissa` and `OutManLen` are to be interpreted as being replaced by the variable names used when the syntax structure is included.

**[00355]** `da_sign_flag` equal to 0 indicates that the sign of the floating-point value is positive. `da_sign_flag` equal to 1 indicates that the sign is negative. The variable `OutSign` is set equal to `da_sign_flag`.

**[00356]** `da_exponent` specifies the exponent of the floating-point value. The value of `da_exponent` shall be in the range of 0 to  $2^7 - 2$ , inclusive. The value  $2^7 - 1$  is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value  $2^7 - 1$  as indicating an unspecified value. The variable `OutExp` is set equal to `da_exponent`.

[00357] **da\_mantissa\_len\_minus1** plus 1 specifies the number of bits in the da\_mantissa syntax element. The value of da\_mantissa\_len\_minus1 shall be in the range of 0 to 31, inclusive. The variable OutManLen is set equal to da\_mantissa\_len\_minus1 + 1.

[00358] **da\_mantissa** specifies the mantissa of the floating-point value. The variable OutMantissa is set equal to da\_mantissa.

[00359] **Embodiment 14**

[00360] Depth representation information SEI message

[00361] Depth representation information SEI message syntax

| depth_representation_info( payloadSize ) {                                 | Descriptor  |
|----------------------------------------------------------------------------|-------------|
| <i>depth_representation_primary_layer_num</i>                              | <i>u(6)</i> |
| <i>for ( i = 0; i &lt; depth_representation_primary_layer_num; i++ ) {</i> |             |
| <i>depth_representation_primary_layer_id[ i ]</i>                          | <i>u(6)</i> |
| <i>}</i>                                                                   |             |
| <b>z_near_flag</b>                                                         | u(1)        |
| <b>z_far_flag</b>                                                          | u(1)        |
| <b>d_min_flag</b>                                                          | u(1)        |
| <b>d_max_flag</b>                                                          | u(1)        |
| <b>depth_representation_type</b>                                           | ue(v)       |
| if( d_min_flag    d_max_flag )                                             |             |
| <b>disparity_ref_view_id</b>                                               | ue(v)       |
| if( z_near_flag )                                                          |             |
| depth_rep_info_element( ZNearSign, ZNearExp, ZNearMantissa, ZNearManLen )  |             |
| if( z_far_flag )                                                           |             |
| depth_rep_info_element( ZFarSign, ZFarExp, ZFarMantissa, ZFarManLen )      |             |
| if( d_min_flag )                                                           |             |
| depth_rep_info_element( DMinSign, DMinExp, DMinMantissa, DMinManLen )      |             |
| if( d_max_flag )                                                           |             |

|                                                                       |       |
|-----------------------------------------------------------------------|-------|
| depth_rep_info_element( DMaxSign, DMaxExp, DMaxMantissa, DMaxManLen ) |       |
| if( depth_representation_type == 3 ) {                                |       |
| <b>depth_nonlinear_representation_num_minus1</b>                      | ue(v) |
| for( i = 1; i <= depth_nonlinear_representation_num_minus1 + 1; i++ ) |       |
| <b>depth_nonlinear_representation_model[ i ]</b>                      |       |
| }                                                                     |       |
| }                                                                     |       |

|                                                                     |                   |
|---------------------------------------------------------------------|-------------------|
| depth_rep_info_element( OutSign, OutExp, OutMantissa, OutManLen ) { | <b>Descriptor</b> |
| <b>da_sign_flag</b>                                                 | u(1)              |
| <b>da_exponent</b>                                                  | u(7)              |
| <b>da_mantissa_len_minus1</b>                                       | u(5)              |
| <b>da_mantissa</b>                                                  | u(v)              |
| }                                                                   |                   |

[00362] Depth representation information SEI message semantics

[00363] The syntax elements in the depth representation information SEI message specify various parameters for auxiliary pictures of type AUX\_DEPTH for the purpose of processing decoded primary and auxiliary pictures prior to rendering on a 3D display, such as view synthesis. Specifically, depth or disparity ranges for depth pictures are specified.

[00364] When present, the depth representation information SEI message shall be associated with one or more layers with sdi\_aux\_id value equal to AUX\_DEPTH. The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the depth representation information SEI message applies.

[00365] When present, the depth representation information SEI message may be included in any access unit. It is recommended that, when present, the SEI message is included for the purpose of random access in an access unit in which the coded picture with nuh\_layer\_id equal to targetLayerId is an IRAP picture.

[00366] For an auxiliary picture with sdi\_aux\_id[ targetLayerId ] equal to AUX\_DEPTH, an associated primary picture, if any, is a picture in the same access unit having

sdi\_aux\_id[ nuhLayerIdB ] equal to 0 such that ScalabilityId[ LayerIdxInVps[ targetLayerId ] ][ j ] is equal to ScalabilityId[ LayerIdxInVps[ nuhLayerIdB ] ][ j ] for all values of j in the range of 0 to 2, inclusive, and 4 to 15, inclusive.

[00367] The information indicated in the SEI message applies to all the pictures with nuh\_layer\_id equal to targetLayerId from the access unit containing the SEI message up to but excluding the next picture, in decoding order, associated with a depth representation information SEI message applicable to targetLayerId or to the end of the CLVS of the nuh\_layer\_id equal to targetLayerId, whichever is earlier in decoding order.

[00368] *depth\_representation\_primary\_layer\_num specifies the number of the associated primary layers to which the depth representation information SEI message applies.*

[00369] *depth\_representation\_primary\_layer\_id[ i ] specifies the nuh\_layer\_id value of the i-th associated primary layer to which the depth representation information SEI message applies.*

[00370] **z\_near\_flag** equal to 0 specifies that the syntax elements specifying the nearest depth value are not present in the syntax structure. **z\_near\_flag** equal to 1 specifies that the syntax elements specifying the nearest depth value are present in the syntax structure.

[00371] **z\_far\_flag** equal to 0 specifies that the syntax elements specifying the farthest depth value are not present in the syntax structure. **z\_far\_flag** equal to 1 specifies that the syntax elements specifying the farthest depth value are present in the syntax structure.

[00372] **d\_min\_flag** equal to 0 specifies that the syntax elements specifying the minimum disparity value are not present in the syntax structure. **d\_min\_flag** equal to 1 specifies that the syntax elements specifying the minimum disparity value are present in the syntax structure.

[00373] **d\_max\_flag** equal to 0 specifies that the syntax elements specifying the maximum disparity value are not present in the syntax structure. **d\_max\_flag** equal to 1 specifies that the syntax elements specifying the maximum disparity value are present in the syntax structure.

[00374] **depth\_representation\_type** specifies the representation definition of decoded luma samples of auxiliary pictures as specified in Table Y1. In Table Y1, disparity specifies the horizontal displacement between two texture views and Z value specifies the distance from a camera.

[00375] The variable maxVal is set equal to  $(1 \ll (8 + \text{sps\_bitdepth\_minus8})) - 1$ , where sps\_bitdepth\_minus8 is the value included in or inferred for the active SPS of the layer with nuh\_layer\_id equal to targetLayerId.

Table Y1 – Definition of depth\_representation\_type

| depth_representation_type | Interpretation                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                         | <p>Each decoded luma sample value of an auxiliary picture represents an inverse of Z value that is uniformly quantized into the range of 0 to maxVal, inclusive.</p> <p>When z_far_flag is equal to 1, the luma sample value equal to 0 represents the inverse of ZFar (specified below). When z_near_flag is equal to 1, the luma sample value equal to maxVal represents the inverse of ZNear (specified below).</p>                                                    |
| 1                         | <p>Each decoded luma sample value of an auxiliary picture represents disparity that is uniformly quantized into the range of 0 to maxVal, inclusive.</p> <p>When d_min_flag is equal to 1, the luma sample value equal to 0 represents DMin (specified below). When d_max_flag is equal to 1, the luma sample value equal to maxVal represents DMax (specified below).</p>                                                                                                |
| 2                         | <p>Each decoded luma sample value of an auxiliary picture represents a Z value uniformly quantized into the range of 0 to maxVal, inclusive.</p> <p>When z_far_flag is equal to 1, the luma sample value equal to 0 corresponds to ZFar (specified below). When z_near_flag is equal to 1, the luma sample value equal to maxVal represents ZNear (specified below).</p>                                                                                                  |
| 3                         | <p>Each decoded luma sample value of an auxiliary picture represents a nonlinearly mapped disparity, normalized in range from 0 to maxVal, as specified by depth_nonlinear_representation_num_minus1 and depth_nonlinear_representation_model[ i ].</p> <p>When d_min_flag is equal to 1, the luma sample value equal to 0 represents DMin (specified below). When d_max_flag is equal to 1, the luma sample value equal to maxVal represents DMax (specified below).</p> |
| Other values              | Reserved for future use                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

[00376] **disparity\_ref\_view\_id** specifies the ViewId value against which the disparity values are derived.

[00377] NOTE 1 – **disparity\_ref\_view\_id** is present only if **d\_min\_flag** is equal to 1 or **d\_max\_flag** is equal to 1 and is useful for **depth\_representation\_type** values equal to 1 and 3.

[00378] The variables in the **x** column of Table Y2 are derived from the respective variables in the **s**, **e**, **n** and **v** columns of Table Y2 as follows:

- If the value of **e** is in the range of 0 to 127, exclusive, **x** is set equal to  $(-1)^s * 2^{e-31} * (1 + n \div 2^v)$ .
- Otherwise (**e** is equal to 0), **x** is set equal to  $(-1)^s * 2^{-(30+v)} * n$ .

[00379] NOTE 1 – The above specification is similar to that found in IEC 60559:1989.

Table Y2 – Association between depth parameter variables and syntax elements

| <b>x</b> | <b>s</b>  | <b>e</b> | <b>n</b>      | <b>v</b>    |
|----------|-----------|----------|---------------|-------------|
| ZNear    | ZNearSign | ZNearExp | ZNearMantissa | ZNearManLen |
| ZFar     | ZFarSign  | ZFarExp  | ZFarMantissa  | ZFarManLen  |
| DMax     | DMaxSign  | DMaxExp  | DMaxMantissa  | DMaxManLen  |
| DMin     | DMinSign  | DMinExp  | DMinMantissa  | DMinManLen  |

[00380] The DMin and DMax values, when present, are specified in units of a luma sample width of the coded picture with ViewId equal to ViewId of the auxiliary picture.

[00381] The units for the ZNear and ZFar values, when present, are identical but unspecified.

[00382] **depth\_nonlinear\_representation\_num\_minus1** plus 2 specifies the number of piece-wise linear segments for mapping of depth values to a scale that is uniformly quantized in terms of disparity.

[00383] **depth\_nonlinear\_representation\_model[i]** for **i** ranging from 0 to **depth\_nonlinear\_representation\_num\_minus1 + 2**, inclusive, specify the piece-wise linear segments for mapping of decoded luma sample values of an auxiliary picture to a scale that is uniformly quantized in terms of disparity. The values of **depth\_nonlinear\_representation\_model[0]** and **depth\_nonlinear\_representation\_model[depth\_nonlinear\_representation\_num\_minus1 + 2]** are both inferred to be equal to 0.

[00384] NOTE 2 – When **depth\_representation\_type** is equal to 3, an auxiliary picture contains nonlinearly transformed depth samples. The variable **DepthLUT[i]**, as specified below,

is used to transform decoded depth sample values from the nonlinear representation to the linear representation, i.e., uniformly quantized disparity values. The shape of this transform is defined by means of line-segment approximation in two-dimensional linear-disparity-to-nonlinear-disparity space. The first (0, 0) and the last (maxVal, maxVal) nodes of the curve are predefined. Positions of additional nodes are transmitted in form of deviations (depth\_nonlinear\_representation\_model[ i ]) from the straight-line curve. These deviations are uniformly distributed along the whole range of 0 to maxVal, inclusive, with spacing depending on the value of nonlinear\_depth\_representation\_num\_minus1.

**[00385]** The variable DepthLUT[ i ] for i in the range of 0 to maxVal, inclusive, is specified as follows:

```
for( k = 0; k <= depth_nonlinear_representation_num_minus1 + 1; k++ ) {
    pos1 = ( maxVal * k ) / (depth_nonlinear_representation_num_minus1 + 2 )
    dev1 = depth_nonlinear_representation_model[ k ]
    pos2 = ( maxVal * ( k + 1 ) ) / (depth_nonlinear_representation_num_minus1 + 2 )
    dev2 = depth_nonlinear_representation_model[ k + 1 ]                (X)

    x1 = pos1 - dev1
    y1 = pos1 + dev1
    x2 = pos2 - dev2
    y2 = pos2 + dev2

    for( x = Max( x1, 0 ); x <= Min( x2, maxVal ); x++ )
        DepthLUT[ x ] = Clip3( 0, maxVal, Round( ( ( x - x1 ) * ( y2 - y1 ) ) ÷ ( x2 - x1 )
+ y1 ) )
}
```

**[00386]** When depth\_representation\_type is equal to 3, DepthLUT[ dS ] for all decoded luma sample values dS of an auxiliary picture in the range of 0 to maxVal, inclusive, represents disparity that is uniformly quantized into the range of 0 to maxVal, inclusive.

**[00387]** The syntax structure specifies the value of an element in the depth representation information SEI message.

**[00388]** The syntax structure sets the values of the OutSign, OutExp, OutMantissa and OutManLen variables that represent a floating-point value. When the syntax structure is included in another syntax structure, the variable names OutSign, OutExp, OutMantissa and OutManLen

are to be interpreted as being replaced by the variable names used when the syntax structure is included.

[00389] **da\_sign\_flag** equal to 0 indicates that the sign of the floating-point value is positive. da\_sign\_flag equal to 1 indicates that the sign is negative. The variable OutSign is set equal to da\_sign\_flag.

[00390] **da\_exponent** specifies the exponent of the floating-point value. The value of da\_exponent shall be in the range of 0 to  $2^7 - 2$ , inclusive. The value  $2^7 - 1$  is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value  $2^7 - 1$  as indicating an unspecified value. The variable OutExp is set equal to da\_exponent.

[00391] **da\_mantissa\_len\_minus1** plus 1 specifies the number of bits in the da\_mantissa syntax element. The value of da\_mantissa\_len\_minus1 shall be in the range of 0 to 31, inclusive. The variable OutManLen is set equal to da\_mantissa\_len\_minus1 + 1.

[00392] **da\_mantissa** specifies the mantissa of the floating-point value. The variable OutMantissa is set equal to da\_mantissa.

[00393] Embodiment 15

[00394] Alpha channel information SEI message

[00395] Alpha channel information SEI message syntax

| alpha_channel_info( payloadSize ) {   | Descriptor  |
|---------------------------------------|-------------|
| <i>alpha_channel_primary_layer_id</i> | <i>u(6)</i> |
| <b>alpha_channel_cancel_flag</b>      | u(1)        |
| if( !alpha_channel_cancel_flag ) {    |             |
| <b>alpha_channel_use_idc</b>          | u(3)        |
| <b>alpha_channel_bit_depth_minus8</b> | u(3)        |
| <b>alpha_transparent_value</b>        | u(v)        |
| <b>alpha_opaque_value</b>             | u(v)        |
| <b>alpha_channel_incr_flag</b>        | u(1)        |
| <b>alpha_channel_clip_flag</b>        | u(1)        |
| if( alpha_channel_clip_flag )         |             |
| <b>alpha_channel_clip_type_flag</b>   | u(1)        |
| }                                     |             |

|   |  |
|---|--|
| } |  |
|---|--|

[00396] Alpha channel information SEI message semantics

[00397] The alpha channel information SEI message provides information about alpha channel sample values and post-processing applied to the decoded alpha planes coded in auxiliary pictures of type AUX\_ALPHA and one or more associated primary pictures.

[00398] For an auxiliary picture with `nuh_layer_id` equal to `nuhLayerIdA` and `sdi_aux_id[ nuhLayerIdA ]` equal to AUX\_ALPHA, an associated primary picture, if any, is a picture in the same access unit having `sdi_aux_id[ nuhLayerIdB ]` equal to 0 such that `ScalabilityId[ LayerIdxInVps[ nuhLayerIdA ] ][ j ]` is equal to `ScalabilityId[ LayerIdxInVps[ nuhLayerIdB ] ][ j ]` for all values of `j` in the range of 0 to 2, inclusive, and 4 to 15, inclusive.

[00399] When an access unit contains an auxiliary picture `picA` with `nuh_layer_id` equal to `nuhLayerIdA` and `sdi_aux_id[ nuhLayerIdA ]` equal to AUX\_ALPHA, the alpha channel sample values of `picA` persist in output order until one or more of the following conditions are true:

- The next picture, in output order, with `nuh_layer_id` equal to `nuhLayerIdA` is output.
- A CLVS containing the auxiliary picture `picA` ends.
- The bitstream ends.
- A CLVS of any associated primary layer of the auxiliary picture layer with `nuh_layer_id` equal to `nuhLayerIdA` ends.

[00400] The following semantics apply separately to each `nuh_layer_id` `targetLayerId` among the `nuh_layer_id` values to which the alpha channel information SEI message applies.

[00401] *alpha\_channel\_primary\_layer\_id specifies the nuh\_layer\_id value of the associated primary layer to which the alpha channel information SEI applies to.*

[00402] `alpha_channel_cancel_flag` equal to 1 indicates that the alpha channel information SEI message cancels the persistence of any previous alpha channel information SEI message in output order that applies to the current layer. `alpha_channel_cancel_flag` equal to 0 indicates that alpha channel information follows.

[00403] Let `currPic` be the picture that the alpha channel information SEI message is associated with. The semantics of alpha channel information SEI message persist for the current layer in output order until one or more of the following conditions are true:

- A new CLVS of the current layer begins.
- The bitstream ends.

– A picture picB with nuh\_layer\_id equal to targetLayerId in an access unit containing an alpha channel information SEI message with nuh\_layer\_id equal to targetLayerId is output having PicOrderCnt( picB ) greater than PicOrderCnt( currPic ), where PicOrderCnt( picB ) and PicOrderCnt( currPic ) are the PicOrderCntVal values of picB and currPic, respectively, immediately after the invocation of the decoding process for picture order count for picB.

**[00404] alpha\_channel\_use\_idc** equal to 0 indicates that for alpha blending purposes the decoded samples of the associated primary picture should be multiplied by the interpretation sample values of the auxiliary coded picture in the display process after output from the decoding process. alpha\_channel\_use\_idc equal to 1 indicates that for alpha blending purposes the decoded samples of the associated primary picture should not be multiplied by the interpretation sample values of the auxiliary coded picture in the display process after output from the decoding process. alpha\_channel\_use\_idc equal to 2 indicates that the usage of the auxiliary picture is unspecified. Values greater than 2 for alpha\_channel\_use\_idc are reserved for future use by ITU-T | ISO/IEC. When not present, the value of alpha\_channel\_use\_idc is inferred to be equal to 2.

**[00405] alpha\_channel\_bit\_depth\_minus8** plus 8 specifies the bit depth of the samples of the luma sample array of the auxiliary picture. alpha\_channel\_bit\_depth\_minus8 shall be in the range 0 to 7 inclusive. alpha\_channel\_bit\_depth\_minus8 shall be equal to bit\_depth\_luma\_minus8 of the associated primary picture.

**[00406] alpha\_transparent\_value** specifies the interpretation sample value of an auxiliary coded picture luma sample for which the associated luma and chroma samples of the primary coded picture are considered transparent for purposes of alpha blending. The number of bits used for the representation of the alpha\_transparent\_value syntax element is alpha\_channel\_bit\_depth\_minus8 + 9.

**[00407] alpha\_opaque\_value** specifies the interpretation sample value of an auxiliary coded picture luma sample for which the associated luma and chroma samples of the primary coded picture are considered opaque for purposes of alpha blending. The number of bits used for the representation of the alpha\_opaque\_value syntax element is alpha\_channel\_bit\_depth\_minus8 + 9.

**[00408] alpha\_channel\_incr\_flag** equal to 0 indicates that the interpretation sample value for each decoded auxiliary picture luma sample value is equal to the decoded auxiliary picture sample value for purposes of alpha blending. alpha\_channel\_incr\_flag equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample value that is greater than Min( alpha\_opaque\_value, alpha\_transparent\_value ) should be increased by one to obtain the interpretation sample value for the auxiliary picture

sample and any auxiliary picture luma sample value that is less than or equal to  $\text{Min}(\text{alpha\_opaque\_value}, \text{alpha\_transparent\_value})$  should be used, without alteration, as the interpretation sample value for the decoded auxiliary picture sample value. When not present, the value of `alpha_channel_incr_flag` is inferred to be equal to 0.

[00409] **alpha\_channel\_clip\_flag** equal to 0 indicates that no clipping operation is applied to obtain the interpretation sample values of the decoded auxiliary picture. `alpha_channel_clip_flag` equal to 1 indicates that the interpretation sample values of the decoded auxiliary picture are altered according to the clipping process described by the `alpha_channel_clip_type_flag` syntax element. When not present, the value of `alpha_channel_clip_flag` is inferred to be equal to 0.

[00410] **alpha\_channel\_clip\_type\_flag** equal to 0 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample that is greater than  $(\text{alpha\_opaque\_value} - \text{alpha\_transparent\_value}) / 2$  is set equal to `alpha_opaque_value` to obtain the interpretation sample value for the auxiliary picture luma sample and any auxiliary picture luma sample that is less or equal than  $(\text{alpha\_opaque\_value} - \text{alpha\_transparent\_value}) / 2$  is set equal to `alpha_transparent_value` to obtain the interpretation sample value for the auxiliary picture luma sample. `alpha_channel_clip_type_flag` equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample that is greater than `alpha_opaque_value` is set equal to `alpha_opaque_value` to obtain the interpretation sample value for the auxiliary picture luma sample and any auxiliary picture luma sample that is less than or equal to `alpha_transparent_value` is set equal to `alpha_transparent_value` to obtain the interpretation sample value for the auxiliary picture luma sample.

[00411] NOTE – When both `alpha_channel_incr_flag` and `alpha_channel_clip_flag` are equal to one, the clipping operation specified by `alpha_channel_clip_type_flag` should be applied first followed by the alteration specified by `alpha_channel_incr_flag` to obtain the interpretation sample value for the auxiliary picture luma sample.

[00412] **Embodiment 16**

[00413] Alpha channel information SEI message

[00414] Alpha channel information SEI message syntax

|                                                            |                                 |
|------------------------------------------------------------|---------------------------------|
| <code>alpha_channel_info( payloadSize ) {</code>           | <b>Descriptor</b>               |
| <b><code>alpha_channel_cancel_flag</code></b>              | <code>u(1)</code>               |
| <code>if( !alpha_channel_cancel_flag ) {</code>            |                                 |
| <b><i><code>alpha_channel_primary_layer_num</code></i></b> | <b><i><code>u(6)</code></i></b> |

|                                                                   |             |
|-------------------------------------------------------------------|-------------|
| <i>for (i = 0; i &lt; alpha_channel_primary_layer_num; i++) {</i> |             |
| <i>    alpha_channel_primary_layer_id</i>                         | <i>u(6)</i> |
| <i>}</i>                                                          |             |
| <b>alpha_channel_use_idc</b>                                      | u(3)        |
| <b>alpha_channel_bit_depth_minus8</b>                             | u(3)        |
| <b>alpha_transparent_value</b>                                    | u(v)        |
| <b>alpha_opaque_value</b>                                         | u(v)        |
| <b>alpha_channel_incr_flag</b>                                    | u(1)        |
| <b>alpha_channel_clip_flag</b>                                    | u(1)        |
| if( alpha_channel_clip_flag )                                     |             |
| <b>alpha_channel_clip_type_flag</b>                               | u(1)        |
| <i>}</i>                                                          |             |
| <i>}</i>                                                          |             |

**[00415]** Alpha channel information SEI message semantics

**[00416]** The alpha channel information SEI message provides information about alpha channel sample values and post-processing applied to the decoded alpha planes coded in auxiliary pictures of type AUX\_ALPHA and one or more associated primary pictures.

**[00417]** For an auxiliary picture with nuh\_layer\_id equal to nuhLayerIdA and sdi\_aux\_id[ nuhLayerIdA ] equal to AUX\_ALPHA, an associated primary picture, if any, is a picture in the same access unit having sdi\_aux\_id[ nuhLayerIdB ] equal to 0 such that ScalabilityId[ LayerIdxInVps[ nuhLayerIdA ] ][ j ] is equal to ScalabilityId[ LayerIdxInVps[ nuhLayerIdB ] ][ j ] for all values of j in the range of 0 to 2, inclusive, and 4 to 15, inclusive.

**[00418]** When an access unit contains an auxiliary picture picA with nuh\_layer\_id equal to nuhLayerIdA and sdi\_aux\_id[ nuhLayerIdA ] equal to AUX\_ALPHA, the alpha channel sample values of picA persist in output order until one or more of the following conditions are true:

- The next picture, in output order, with nuh\_layer\_id equal to nuhLayerIdA is output.
- A CLVS containing the auxiliary picture picA ends.
- The bitstream ends.

– A CLVS of any associated primary layer of the auxiliary picture layer with nuh\_layer\_id equal to nuhLayerIdA ends.

[00419] The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the alpha channel information SEI message applies.

[00420] **alpha\_channel\_cancel\_flag** equal to 1 indicates that the alpha channel information SEI message cancels the persistence of any previous alpha channel information SEI message in output order that applies to the current layer. alpha\_channel\_cancel\_flag equal to 0 indicates that alpha channel information follows.

[00421] Let currPic be the picture that the alpha channel information SEI message is associated with. The semantics of alpha channel information SEI message persist for the current layer in output order until one or more of the following conditions are true:

- A new CLVS of the current layer begins.
- The bitstream ends.
- A picture picB with nuh\_layer\_id equal to targetLayerId in an access unit containing an alpha channel information SEI message with nuh\_layer\_id equal to targetLayerId is output having PicOrderCnt( picB ) greater than PicOrderCnt( currPic ), where PicOrderCnt( picB ) and PicOrderCnt( currPic ) are the PicOrderCntVal values of picB and currPic, respectively, immediately after the invocation of the decoding process for picture order count for picB.

[00422] *alpha\_channel\_primary\_layer\_num specifies the number of the associated primary layers to which the alpha channel information SEI message applies.*

[00423] *alpha\_channel\_primary\_layer\_id[ i ] specifies the nuh\_layer\_id value of the i-th associated primary layer to which the alpha channel information SEI message applies.*

[00424] **alpha\_channel\_use\_idc** equal to 0 indicates that for alpha blending purposes the decoded samples of the associated primary picture should be multiplied by the interpretation sample values of the auxiliary coded picture in the display process after output from the decoding process. alpha\_channel\_use\_idc equal to 1 indicates that for alpha blending purposes the decoded samples of the associated primary picture should not be multiplied by the interpretation sample values of the auxiliary coded picture in the display process after output from the decoding process. alpha\_channel\_use\_idc equal to 2 indicates that the usage of the auxiliary picture is unspecified. Values greater than 2 for alpha\_channel\_use\_idc are reserved for future use by ITU-T | ISO/IEC. When not present, the value of alpha\_channel\_use\_idc is inferred to be equal to 2.

[00425] **alpha\_channel\_bit\_depth\_minus8** plus 8 specifies the bit depth of the samples of the luma sample array of the auxiliary picture. alpha\_channel\_bit\_depth\_minus8 shall be in the

range 0 to 7 inclusive. `alpha_channel_bit_depth_minus8` shall be equal to `bit_depth_luma_minus8` of the associated primary picture.

**[00426] `alpha_transparent_value`** specifies the interpretation sample value of an auxiliary coded picture luma sample for which the associated luma and chroma samples of the primary coded picture are considered transparent for purposes of alpha blending. The number of bits used for the representation of the `alpha_transparent_value` syntax element is `alpha_channel_bit_depth_minus8 + 9`.

**[00427] `alpha_opaque_value`** specifies the interpretation sample value of an auxiliary coded picture luma sample for which the associated luma and chroma samples of the primary coded picture are considered opaque for purposes of alpha blending. The number of bits used for the representation of the `alpha_opaque_value` syntax element is `alpha_channel_bit_depth_minus8 + 9`.

**[00428] `alpha_channel_incr_flag`** equal to 0 indicates that the interpretation sample value for each decoded auxiliary picture luma sample value is equal to the decoded auxiliary picture sample value for purposes of alpha blending. `alpha_channel_incr_flag` equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample value that is greater than  $\text{Min}(\text{alpha\_opaque\_value}, \text{alpha\_transparent\_value})$  should be increased by one to obtain the interpretation sample value for the auxiliary picture sample and any auxiliary picture luma sample value that is less than or equal to  $\text{Min}(\text{alpha\_opaque\_value}, \text{alpha\_transparent\_value})$  should be used, without alteration, as the interpretation sample value for the decoded auxiliary picture sample value. When not present, the value of `alpha_channel_incr_flag` is inferred to be equal to 0.

**[00429] `alpha_channel_clip_flag`** equal to 0 indicates that no clipping operation is applied to obtain the interpretation sample values of the decoded auxiliary picture. `alpha_channel_clip_flag` equal to 1 indicates that the interpretation sample values of the decoded auxiliary picture are altered according to the clipping process described by the `alpha_channel_clip_type_flag` syntax element. When not present, the value of `alpha_channel_clip_flag` is inferred to be equal to 0.

**[00430] `alpha_channel_clip_type_flag`** equal to 0 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample that is greater than  $(\text{alpha\_opaque\_value} - \text{alpha\_transparent\_value}) / 2$  is set equal to `alpha_opaque_value` to obtain the interpretation sample value for the auxiliary picture luma sample and any auxiliary picture luma sample that is less or equal than  $(\text{alpha\_opaque\_value} - \text{alpha\_transparent\_value}) / 2$  is set equal to `alpha_transparent_value` to obtain the interpretation sample value for the auxiliary picture luma sample.

alpha\_channel\_clip\_type\_flag equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample that is greater than alpha\_opaque\_value is set equal to alpha\_opaque\_value to obtain the interpretation sample value for the auxiliary picture luma sample and any auxiliary picture luma sample that is less than or equal to alpha\_transparent\_value is set equal to alpha\_transparent\_value to obtain the interpretation sample value for the auxiliary picture luma sample.

[00431] NOTE – When both alpha\_channel\_incr\_flag and alpha\_channel\_clip\_flag are equal to one, the clipping operation specified by alpha\_channel\_clip\_type\_flag should be applied first followed by the alteration specified by alpha\_channel\_incr\_flag to obtain the interpretation sample value for the auxiliary picture luma sample.

[00432] Embodiment 17

[00433] Multiview acquisition information SEI message

[00434] Multiview acquisition information SEI message syntax

| multiview_acquisition_info( payloadSize ) {                                   | Descriptor |
|-------------------------------------------------------------------------------|------------|
| <b>intrinsic_param_flag</b>                                                   | u(1)       |
| <b>extrinsic_param_flag</b>                                                   | u(1)       |
| if( intrinsic_param_flag ) {                                                  |            |
| <b>intrinsic_params_equal_flag</b>                                            | u(1)       |
| <b>prec_focal_length</b>                                                      | ue(v)      |
| <b>prec_principal_point</b>                                                   | ue(v)      |
| <b>prec_skew_factor</b>                                                       | ue(v)      |
| for( i = 0; i <= intrinsic_params_equal_flag ? 0 : numViewsMinus1; i++ )<br>{ |            |
| <b>sign_focal_length_x[ i ]</b>                                               | u(1)       |
| <b>exponent_focal_length_x[ i ]</b>                                           | u(6)       |
| <b>mantissa_focal_length_x[ i ]</b>                                           | u(v)       |
| <b>sign_focal_length_y[ i ]</b>                                               | u(1)       |
| <b>exponent_focal_length_y[ i ]</b>                                           | u(6)       |
| <b>mantissa_focal_length_y[ i ]</b>                                           | u(v)       |
| <b>sign_principal_point_x[ i ]</b>                                            | u(1)       |

|                                         |       |
|-----------------------------------------|-------|
| <b>exponent_principal_point_x[ i ]</b>  | u(6)  |
| <b>mantissa_principal_point_x[ i ]</b>  | u(v)  |
| <b>sign_principal_point_y[ i ]</b>      | u(1)  |
| <b>exponent_principal_point_y[ i ]</b>  | u(6)  |
| <b>mantissa_principal_point_y[ i ]</b>  | u(v)  |
| <b>sign_skew_factor[ i ]</b>            | u(1)  |
| <b>exponent_skew_factor[ i ]</b>        | u(6)  |
| <b>mantissa_skew_factor[ i ]</b>        | u(v)  |
| }                                       |       |
| }                                       |       |
| if( extrinsic_param_flag ) {            |       |
| <b>prec_rotation_param</b>              | ue(v) |
| <b>prec_translation_param</b>           | ue(v) |
| for( i = 0; i <= numViewsMinus1; i++ )  |       |
| for( j = 0; j < 3; j++ ) { /* row */    |       |
| for( k = 0; k < 3; k++ ) { /* column */ |       |
| <b>sign_r[ i ][ j ][ k ]</b>            | u(1)  |
| <b>exponent_r[ i ][ j ][ k ]</b>        | u(6)  |
| <b>mantissa_r[ i ][ j ][ k ]</b>        | u(v)  |
| }                                       |       |
| <b>sign_t[ i ][ j ]</b>                 | u(1)  |
| <b>exponent_t[ i ][ j ]</b>             | u(6)  |
| <b>mantissa_t[ i ][ j ]</b>             | u(v)  |
| }                                       |       |
| }                                       |       |
| }                                       |       |

[00435] Multiview acquisition information SEI message semantics

[00436] The multiview acquisition information (*MAI*) SEI message specifies various parameters of the acquisition environment. Specifically, intrinsic and extrinsic camera parameters are specified. These parameters could be used for processing the decoded views prior to rendering on a 3D display.

[00437] The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the multiview acquisition information SEI message applies.

[00438] When present, the multiview acquisition information SEI message that applies to the current layer shall be included in an access unit that contains an IRAP picture that is the first picture of a CLVS of the current layer. The information signalled in the SEI message applies to the CLVS.

[00439] *An MAI SEI message that has payloadType equal to 179 (multiview acquisition) shall not be contained in a scalable nesting SEI message.*

[00440] *Let the current AU be the AU containing the current MAI SEI message, and the current CVS be the CVS containing the current AU.*

[00441] *When a CVS does not contain an SDI SEI message, the CVS shall not contain an MAI SEI message.*

[00442] *When an AU contains both an SDI SEI message and an MAI SEI message, the SDI SEI message shall precede the MAI SEI message in decoding order.*

[00443] When the multiview acquisition information SEI message is contained in a scalable nesting SEI message, the syntax elements sn\_ols\_flag and sn\_all\_layers\_flag in the scalable nesting SEI message shall be equal to 0.

[00444] The variable numViewsMinus1 is derived as follows:

- If the multiview acquisition information SEI message is not included in a scalable nesting SEI message, numViewsMinus1 is set equal to 0.
- Otherwise (the multiview acquisition information SEI message is included in a scalable nesting SEI message), numViewsMinus1 is set equal to sn\_num\_layers\_minus1.

[00445] Some of the views for which the multiview acquisition information is included in a multiview acquisition information SEI message may not be present.

[00446] In the semantics below, index *i* refers to the syntax elements and variables that apply to the layer with nuh\_layer\_id equal to NestingLayerId[ *i* ].

[00447] The extrinsic camera parameters are specified according to a right-handed coordinate system, where the upper left corner of the image is the origin, i.e., the ( 0, 0 ) coordinate, with the other corners of the image having non-negative coordinates. With these specifications, a 3-

dimensional world point,  $wP = [x\ y\ z]$  is mapped to a 2-dimensional camera point,  $cP[i] = [u\ v\ 1]$ , for the  $i$ -th camera according to:

$$s * cP[i] = A[i] * R^{-1}[i] * (wP - T[i]) \quad (X)$$

[00448] where  $A[i]$  denotes the intrinsic camera parameter matrix,  $R^{-1}[i]$  denotes the inverse of the rotation matrix  $R[i]$ ,  $T[i]$  denotes the translation vector and  $s$  (a scalar value) is an arbitrary scale factor chosen to make the third coordinate of  $cP[i]$  equal to 1. The elements of  $A[i]$ ,  $R[i]$  and  $T[i]$  are determined according to the syntax elements signalled in this SEI message and as specified below.

[00449] **intrinsic\_param\_flag** equal to 1 indicates the presence of intrinsic camera parameters. **intrinsic\_param\_flag** equal to 0 indicates the absence of intrinsic camera parameters.

[00450] **extrinsic\_param\_flag** equal to 1 indicates the presence of extrinsic camera parameters. **extrinsic\_param\_flag** equal to 0 indicates the absence of extrinsic camera parameters.

[00451] **intrinsic\_params\_equal\_flag** equal to 1 indicates that the intrinsic camera parameters are equal for all cameras and only one set of intrinsic camera parameters are present. **intrinsic\_params\_equal\_flag** equal to 0 indicates that the intrinsic camera parameters are different for each camera and that a set of intrinsic camera parameters are present for each camera.

[00452] **prec\_focal\_length** specifies the exponent of the maximum allowable truncation error for **focal\_length\_x[i]** and **focal\_length\_y[i]** as given by  $2^{-\text{prec\_focal\_length}}$ . The value of **prec\_focal\_length** shall be in the range of 0 to 31, inclusive.

[00453] **prec\_principal\_point** specifies the exponent of the maximum allowable truncation error for **principal\_point\_x[i]** and **principal\_point\_y[i]** as given by  $2^{-\text{prec\_principal\_point}}$ . The value of **prec\_principal\_point** shall be in the range of 0 to 31, inclusive.

[00454] **prec\_skew\_factor** specifies the exponent of the maximum allowable truncation error for skew factor as given by  $2^{-\text{prec\_skew\_factor}}$ . The value of **prec\_skew\_factor** shall be in the range of 0 to 31, inclusive.

[00455] **sign\_focal\_length\_x[i]** equal to 0 indicates that the sign of the focal length of the  $i$ -th camera in the horizontal direction is positive. **sign\_focal\_length\_x[i]** equal to 1 indicates that the sign is negative.

[00456] **exponent\_focal\_length\_x[i]** specifies the exponent part of the focal length of the  $i$ -th camera in the horizontal direction. The value of **exponent\_focal\_length\_x[i]** shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified focal length.

[00457] **mantissa\_focal\_length\_x[ i ]** specifies the mantissa part of the focal length of the i-th camera in the horizontal direction. The length of the mantissa\_focal\_length\_x[ i ] syntax element is variable and determined as follows:

- If exponent\_focal\_length\_x[ i ] is equal to 0, the length is  $\text{Max}(0, \text{prec\_focal\_length} - 30)$ .
- Otherwise (exponent\_focal\_length\_x[ i ] is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_focal\_length\_x}[ i ] + \text{prec\_focal\_length} - 31)$ .

**sign\_focal\_length\_y[ i ]** equal to 0 indicates that the sign of the focal length of the i-th camera in the vertical direction is positive. **sign\_focal\_length\_y[ i ]** equal to 1 indicates that the sign is negative.

[00458] **exponent\_focal\_length\_y[ i ]** specifies the exponent part of the focal length of the i-th camera in the vertical direction. The value of exponent\_focal\_length\_y[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified focal length.

[00459] **mantissa\_focal\_length\_y[ i ]** specifies the mantissa part of the focal length of the i-th camera in the vertical direction.

[00460] The length of the mantissa\_focal\_length\_y[ i ] syntax element is variable and determined as follows:

- If exponent\_focal\_length\_y[ i ] is equal to 0, the length is  $\text{Max}(0, \text{prec\_focal\_length} - 30)$ .
- Otherwise (exponent\_focal\_length\_y[ i ] is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_focal\_length\_y}[ i ] + \text{prec\_focal\_length} - 31)$ .

[00461] **sign\_principal\_point\_x[ i ]** equal to 0 indicates that the sign of the principal point of the i-th camera in the horizontal direction is positive. **sign\_principal\_point\_x[ i ]** equal to 1 indicates that the sign is negative.

[00462] **exponent\_principal\_point\_x[ i ]** specifies the exponent part of the principal point of the i-th camera in the horizontal direction. The value of exponent\_principal\_point\_x[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified principal point.

[00463] **mantissa\_principal\_point\_x[ i ]** specifies the mantissa part of the principal point of the i-th camera in the horizontal direction. The length of the mantissa\_principal\_point\_x[ i ] syntax element in units of bits is variable and is determined as follows:

- If exponent\_principal\_point\_x[ i ] is equal to 0, the length is  $\text{Max}(0, \text{prec\_principal\_point} - 30)$ .

– Otherwise ( $\text{exponent\_principal\_point\_x}[i]$  is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_principal\_point\_x}[i] + \text{prec\_principal\_point} - 31)$ .

**[00464] sign\_principal\_point\_y[i]** equal to 0 indicates that the sign of the principal point of the i-th camera in the vertical direction is positive.  $\text{sign\_principal\_point\_y}[i]$  equal to 1 indicates that the sign is negative.

**[00465] exponent\_principal\_point\_y[i]** specifies the exponent part of the principal point of the i-th camera in the vertical direction. The value of  $\text{exponent\_principal\_point\_y}[i]$  shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified principal point.

**[00466] mantissa\_principal\_point\_y[i]** specifies the mantissa part of the principal point of the i-th camera in the vertical direction. The length of the  $\text{mantissa\_principal\_point\_y}[i]$  syntax element in units of bits is variable and is determined as follows:

– If  $\text{exponent\_principal\_point\_y}[i]$  is equal to 0, the length is  $\text{Max}(0, \text{prec\_principal\_point} - 30)$ .

– Otherwise ( $\text{exponent\_principal\_point\_y}[i]$  is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_principal\_point\_y}[i] + \text{prec\_principal\_point} - 31)$ .

**[00467] sign\_skew\_factor[i]** equal to 0 indicates that the sign of the skew factor of the i-th camera is positive.

**[00468] sign\_skew\_factor[i]** equal to 1 indicates that the sign is negative.

**[00469] exponent\_skew\_factor[i]** specifies the exponent part of the skew factor of the i-th camera. The value of  $\text{exponent\_skew\_factor}[i]$  shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified skew factor.

**[00470] mantissa\_skew\_factor[i]** specifies the mantissa part of the skew factor of the i-th camera. The length of the  $\text{mantissa\_skew\_factor}[i]$  syntax element is variable and determined as follows:

– If  $\text{exponent\_skew\_factor}[i]$  is equal to 0, the length is  $\text{Max}(0, \text{prec\_skew\_factor} - 30)$ .

– Otherwise ( $\text{exponent\_skew\_factor}[i]$  is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_skew\_factor}[i] + \text{prec\_skew\_factor} - 31)$ .

**[00471]** The intrinsic matrix  $A[i]$  for i-th camera is represented by

$$\begin{bmatrix} \text{focalLengthX}[i] & \text{skewFactor}[i] & \text{principalPointX}[i] \\ 0 & \text{focalLengthY}[i] & \text{principalPointY}[i] \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{X})$$

[00472] **prec\_rotation\_param** specifies the exponent of the maximum allowable truncation error for  $r[i][j][k]$  as given by  $2^{-\text{prec\_rotation\_param}}$ . The value of **prec\_rotation\_param** shall be in the range of 0 to 31, inclusive.

[00473] **prec\_translation\_param** specifies the exponent of the maximum allowable truncation error for  $t[i][j]$  as given by  $2^{-\text{prec\_translation\_param}}$ . The value of **prec\_translation\_param** shall be in the range of 0 to 31, inclusive.

[00474] **sign\_r**[ $i][j][k]$  equal to 0 indicates that the sign of  $(j, k)$  component of the rotation matrix for the  $i$ -th camera is positive. **sign\_r**[ $i][j][k]$  equal to 1 indicates that the sign is negative.

[00475] **exponent\_r**[ $i][j][k]$  specifies the exponent part of  $(j, k)$  component of the rotation matrix for the  $i$ -th camera. The value of **exponent\_r**[ $i][j][k]$  shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified rotation matrix.

[00476] **mantissa\_r**[ $i][j][k]$  specifies the mantissa part of  $(j, k)$  component of the rotation matrix for the  $i$ -th camera. The length of the **mantissa\_r**[ $i][j][k]$  syntax element in units of bits is variable and determined as follows:

- If **exponent\_r**[ $i]$  is equal to 0, the length is  $\text{Max}(0, \text{prec\_rotation\_param} - 30)$ .
- Otherwise (**exponent\_r**[ $i]$  is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_r}[i] + \text{prec\_rotation\_param} - 31)$ .

[00477] The rotation matrix  $R[i]$  for  $i$ -th camera is represented as follows:

$$\begin{bmatrix} rE[i][0][0] & rE[i][0][1] & rE[i][0][2] \\ rE[i][1][0] & rE[i][1][1] & rE[i][1][2] \\ rE[i][2][0] & rE[i][2][1] & rE[i][2][2] \end{bmatrix} \quad (\text{X})$$

[00478] **sign\_t**[ $i][j]$  equal to 0 indicates that the sign of the  $j$ -th component of the translation vector for the  $i$ -th camera is positive. **sign\_t**[ $i][j]$  equal to 1 indicates that the sign is negative.

[00479] **exponent\_t**[ $i][j]$  specifies the exponent part of the  $j$ -th component of the translation vector for the  $i$ -th camera. The value of **exponent\_t**[ $i][j]$  shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified translation vector.

[00480] **mantissa\_t[ i ][ j ]** specifies the mantissa part of the j-th component of the translation vector for the i-th camera. The length v of the mantissa\_t[ i ][ j ] syntax element in units of bits is variable and is determined as follows:

- If exponent\_t[ i ] is equal to 0, the length v is set equal to  $\text{Max}(0, \text{prec\_translation\_param} - 30)$ .
- Otherwise ( $0 < \text{exponent\_t}[ i ] < 63$ ), the length v is set equal to  $\text{Max}(0, \text{exponent\_t}[ i ] + \text{prec\_translation\_param} - 31)$ .

[00481] The translation vector  $T[ i ]$  for the i-th camera is represented by:

$$\begin{bmatrix} tE[i][0] \\ tE[i][1] \\ tE[i][2] \end{bmatrix} \quad (X)$$

[00482] The association between the camera parameter variables and corresponding syntax elements is specified by Table ZZ. Each component of the intrinsic and rotation matrices and the translation vector is obtained from the variables specified in Table ZZ as the variable x computed as follows:

- If e is in the range of 0 to 63, exclusive, x is set equal to  $(-1)^s * 2^{e-31} * (1 + n \div 2^v)$ .
- Otherwise (e is equal to 0), x is set equal to  $(-1)^s * 2^{-(30+v)} * n$ .

[00483] NOTE – The above specification is similar to that found in IEC 60559:1989.

Table ZZ – Association between camera parameter variables and syntax elements.

| <b>x</b>                     | <b>s</b>                    | <b>e</b>                        | <b>n</b>                        |
|------------------------------|-----------------------------|---------------------------------|---------------------------------|
| <b>focalLengthX[ i ]</b>     | sign_focal_length_x[ i ]    | exponent_focal_length_x[ i ]    | mantissa_focal_length_x[ i ]    |
| <b>focalLengthY[ i ]</b>     | sign_focal_length_y[ i ]    | exponent_focal_length_y[ i ]    | mantissa_focal_length_y[ i ]    |
| <b>principalPoint X[ i ]</b> | sign_principal_point_x[ i ] | exponent_principal_point_x[ i ] | mantissa_principal_point_x[ i ] |
| <b>principalPoint Y[ i ]</b> | sign_principal_point_y[ i ] | exponent_principal_point_y[ i ] | mantissa_principal_point_y[ i ] |
| <b>skewFactor[ i ]</b>       | sign_skew_factor[ i ]       | exponent_skew_factor[ i ]       | mantissa_skew_factor[ i ]       |
| <b>rE[ i ][ j ][ k ]</b>     | sign_r[ i ][ j ][ k ]       | exponent_r[ i ][ j ][ k ]       | mantissa_r[ i ][ j ][ k ]       |

|                      |                  |                      |                      |
|----------------------|------------------|----------------------|----------------------|
| <b>tE</b> [ i ][ j ] | sign_t[ i ][ j ] | exponent_t[ i ][ j ] | mantissa_t[ i ][ j ] |
|----------------------|------------------|----------------------|----------------------|

**[00484] Embodiment 18****[00485]** Depth representation information SEI message**[00486]** Depth representation information SEI message syntax

| depth_representation_info( payloadSize ) {                                | Descriptor |
|---------------------------------------------------------------------------|------------|
| <b>z_near_flag</b>                                                        | u(1)       |
| <b>z_far_flag</b>                                                         | u(1)       |
| <b>d_min_flag</b>                                                         | u(1)       |
| <b>d_max_flag</b>                                                         | u(1)       |
| <b>depth_representation_type</b>                                          | ue(v)      |
| if( d_min_flag    d_max_flag )                                            |            |
| <b>disparity_ref_view_id</b>                                              | ue(v)      |
| if( z_near_flag )                                                         |            |
| depth_rep_info_element( ZNearSign, ZNearExp, ZNearMantissa, ZNearManLen ) |            |
| if( z_far_flag )                                                          |            |
| depth_rep_info_element( ZFarSign, ZFarExp, ZFarMantissa, ZFarManLen )     |            |
| if( d_min_flag )                                                          |            |
| depth_rep_info_element( DMinSign, DMinExp, DMinMantissa, DMinManLen )     |            |
| if( d_max_flag )                                                          |            |
| depth_rep_info_element( DMaxSign, DMaxExp, DMaxMantissa, DMaxManLen )     |            |
| if( depth_representation_type == 3 ) {                                    |            |
| <b>depth_nonlinear_representation_num_minus1</b>                          | ue(v)      |
| for( i = 1; i <= depth_nonlinear_representation_num_minus1 + 1; i++ )     |            |
| <b>depth_nonlinear_representation_model</b> [ i ]                         |            |
| }                                                                         |            |

|   |  |
|---|--|
| } |  |
|---|--|

|                                                                     |                   |
|---------------------------------------------------------------------|-------------------|
| depth_rep_info_element( OutSign, OutExp, OutMantissa, OutManLen ) { | <b>Descriptor</b> |
| da_sign_flag                                                        | u(1)              |
| da_exponent                                                         | u(7)              |
| da_mantissa_len_minus1                                              | u(5)              |
| da_mantissa                                                         | u(v)              |
| }                                                                   |                   |

[00487] Depth representation information SEI message semantics

[00488] The syntax elements in the depth representation information SEI message specify various parameters for auxiliary pictures of type AUX\_DEPTH for the purpose of processing decoded primary and auxiliary pictures prior to rendering on a 3D display, such as view synthesis. Specifically, depth or disparity ranges for depth pictures are specified.

[00489] When present, the depth representation information SEI message shall be associated with one or more layers with sdi\_aux\_id value equal to AUX\_DEPTH. The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the depth representation information SEI message applies.

[00490] When present, the depth representation information SEI message may be included in any access unit. It is recommended that, when present, the SEI message is included for the purpose of random access in an access unit in which the coded picture with nuh\_layer\_id equal to targetLayerId is an IRAP picture.

[00491] *It is a requirement of bitstream conformance that the depth representation information SEI message shall not be present in the bitstream in which the scalability dimension information SEI message is not present.*

[00492] For an auxiliary picture with sdi\_aux\_id[ targetLayerId ] equal to AUX\_DEPTH, an associated primary picture, if any, is a picture in the same access unit having sdi\_aux\_id[ nuhLayerIdB ] equal to 0 such that ScalabilityId[ LayerIdxInVps[ targetLayerId ] ][ j ] is equal to ScalabilityId[ LayerIdxInVps[ nuhLayerIdB ] ][ j ] for all values of j in the range of 0 to 2, inclusive, and 4 to 15, inclusive.

[00493] The information indicated in the SEI message applies to all the pictures with nuh\_layer\_id equal to targetLayerId from the access unit containing the SEI message up to but excluding the next picture, in decoding order, associated with a depth representation information SEI message applicable to targetLayerId or to the end of the CLVS of the nuh\_layer\_id equal to targetLayerId, whichever is earlier in decoding order.

[00494] **z\_near\_flag** equal to 0 specifies that the syntax elements specifying the nearest depth value are not present in the syntax structure. z\_near\_flag equal to 1 specifies that the syntax elements specifying the nearest depth value are present in the syntax structure.

[00495] **z\_far\_flag** equal to 0 specifies that the syntax elements specifying the farthest depth value are not present in the syntax structure. z\_far\_flag equal to 1 specifies that the syntax elements specifying the farthest depth value are present in the syntax structure.

[00496] **d\_min\_flag** equal to 0 specifies that the syntax elements specifying the minimum disparity value are not present in the syntax structure. d\_min\_flag equal to 1 specifies that the syntax elements specifying the minimum disparity value are present in the syntax structure.

[00497] **d\_max\_flag** equal to 0 specifies that the syntax elements specifying the maximum disparity value are not present in the syntax structure. d\_max\_flag equal to 1 specifies that the syntax elements specifying the maximum disparity value are present in the syntax structure.

[00498] **depth\_representation\_type** specifies the representation definition of decoded luma samples of auxiliary pictures as specified in Table Y1. In Table Y1, disparity specifies the horizontal displacement between two texture views and Z value specifies the distance from a camera.

[00499] The variable maxVal is set equal to  $(1 \ll (8 + \text{sps\_bitdepth\_minus8})) - 1$ , where sps\_bitdepth\_minus8 is the value included in or inferred for the active SPS of the layer with nuh\_layer\_id equal to targetLayerId.

Table Y1 – Definition of depth\_representation\_type

| depth_representation_type | Interpretation                                                                                                                                                |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                         | Each decoded luma sample value of an auxiliary picture represents an inverse of Z value that is uniformly quantized into the range of 0 to maxVal, inclusive. |

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | When <code>z_far_flag</code> is equal to 1, the luma sample value equal to 0 represents the inverse of <code>ZFar</code> (specified below). When <code>z_near_flag</code> is equal to 1, the luma sample value equal to <code>maxVal</code> represents the inverse of <code>ZNear</code> (specified below).                                                                                                                                                                                                                                                                |
| 1            | Each decoded luma sample value of an auxiliary picture represents disparity that is uniformly quantized into the range of 0 to <code>maxVal</code> , inclusive.<br><br>When <code>d_min_flag</code> is equal to 1, the luma sample value equal to 0 represents <code>DMin</code> (specified below). When <code>d_max_flag</code> is equal to 1, the luma sample value equal to <code>maxVal</code> represents <code>DMax</code> (specified below).                                                                                                                         |
| 2            | Each decoded luma sample value of an auxiliary picture represents a <code>Z</code> value uniformly quantized into the range of 0 to <code>maxVal</code> , inclusive.<br><br>When <code>z_far_flag</code> is equal to 1, the luma sample value equal to 0 corresponds to <code>ZFar</code> (specified below). When <code>z_near_flag</code> is equal to 1, the luma sample value equal to <code>maxVal</code> represents <code>ZNear</code> (specified below).                                                                                                              |
| 3            | Each decoded luma sample value of an auxiliary picture represents a nonlinearly mapped disparity, normalized in range from 0 to <code>maxVal</code> , as specified by <code>depth_nonlinear_representation_num_minus1</code> and <code>depth_nonlinear_representation_model[i]</code> .<br><br>When <code>d_min_flag</code> is equal to 1, the luma sample value equal to 0 represents <code>DMin</code> (specified below). When <code>d_max_flag</code> is equal to 1, the luma sample value equal to <code>maxVal</code> represents <code>DMax</code> (specified below). |
| Other values | Reserved for future use                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

**[00500]** `disparity_ref_view_id` specifies the `ViewId` value against which the disparity values are derived.

**[00501]** NOTE 1 – `disparity_ref_view_id` is present only if `d_min_flag` is equal to 1 or `d_max_flag` is equal to 1 and is useful for `depth_representation_type` values equal to 1 and 3.

[00502] The variables in the x column of Table Y2 are derived from the respective variables in the s, e, n and v columns of Table Y2 as follows:

- If the value of e is in the range of 0 to 127, exclusive, x is set equal to  $(-1)^s * 2^{e-31} * (1 + n \div 2^v)$ .
- Otherwise (e is equal to 0), x is set equal to  $(-1)^s * 2^{-(30+v)} * n$ .

[00503] NOTE 1 – The above specification is similar to that found in IEC 60559:1989.

Table Y2 – Association between depth parameter variables and syntax elements

| x     | s         | e        | n             | v           |
|-------|-----------|----------|---------------|-------------|
| ZNear | ZNearSign | ZNearExp | ZNearMantissa | ZNearManLen |
| ZFar  | ZFarSign  | ZFarExp  | ZFarMantissa  | ZFarManLen  |
| DMax  | DMaxSign  | DMaxExp  | DMaxMantissa  | DMaxManLen  |
| DMin  | DMinSign  | DMinExp  | DMinMantissa  | DMinManLen  |

[00504] The DMin and DMax values, when present, are specified in units of a luma sample width of the coded picture with ViewId equal to ViewId of the auxiliary picture.

[00505] The units for the ZNear and ZFar values, when present, are identical but unspecified.

[00506] **depth\_nonlinear\_representation\_num\_minus1** plus 2 specifies the number of piece-wise linear segments for mapping of depth values to a scale that is uniformly quantized in terms of disparity.

[00507] **depth\_nonlinear\_representation\_model[i]** for i ranging from 0 to **depth\_nonlinear\_representation\_num\_minus1** + 2, inclusive, specify the piece-wise linear segments for mapping of decoded luma sample values of an auxiliary picture to a scale that is uniformly quantized in terms of disparity. The values of **depth\_nonlinear\_representation\_model[0]** and **depth\_nonlinear\_representation\_model[depth\_nonlinear\_representation\_num\_minus1 + 2]** are both inferred to be equal to 0.

[00508] NOTE 2 – When **depth\_representation\_type** is equal to 3, an auxiliary picture contains nonlinearly transformed depth samples. The variable **DepthLUT[i]**, as specified below, is used to transform decoded depth sample values from the nonlinear representation to the linear representation, i.e., uniformly quantized disparity values. The shape of this transform is defined by means of line-segment approximation in two-dimensional linear-disparity-to-nonlinear-disparity space. The first (0, 0) and the last (maxVal, maxVal) nodes of the curve are

predefined. Positions of additional nodes are transmitted in form of deviations ( $\text{depth\_nonlinear\_representation\_model}[i]$ ) from the straight-line curve. These deviations are uniformly distributed along the whole range of 0 to  $\text{maxVal}$ , inclusive, with spacing depending on the value of  $\text{nonlinear\_depth\_representation\_num\_minus1}$ .

**[00509]** The variable  $\text{DepthLUT}[i]$  for  $i$  in the range of 0 to  $\text{maxVal}$ , inclusive, is specified as follows:

```
for( k = 0; k <= depth_nonlinear_representation_num_minus1 + 1; k++ ) {
    pos1 = ( maxVal * k ) / ( depth_nonlinear_representation_num_minus1 + 2 )
    dev1 = depth_nonlinear_representation_model[ k ]
    pos2 = ( maxVal * ( k + 1 ) ) / ( depth_nonlinear_representation_num_minus1 + 2 )
    dev2 = depth_nonlinear_representation_model[ k + 1 ]   (X)

    x1 = pos1 - dev1
    y1 = pos1 + dev1
    x2 = pos2 - dev2
    y2 = pos2 + dev2

    for( x = Max( x1, 0 ); x <= Min( x2, maxVal ); x++ )
        DepthLUT[ x ] = Clip3( 0, maxVal, Round( ( ( x - x1 ) * ( y2 - y1 ) ) ÷
            ( x2 - x1 ) + y1 ) )
}
```

**[00510]** When  $\text{depth\_representation\_type}$  is equal to 3,  $\text{DepthLUT}[dS]$  for all decoded luma sample values  $dS$  of an auxiliary picture in the range of 0 to  $\text{maxVal}$ , inclusive, represents disparity that is uniformly quantized into the range of 0 to  $\text{maxVal}$ , inclusive.

**[00511]** The syntax structure specifies the value of an element in the depth representation information SEI message.

**[00512]** The syntax structure sets the values of the  $\text{OutSign}$ ,  $\text{OutExp}$ ,  $\text{OutMantissa}$  and  $\text{OutManLen}$  variables that represent a floating-point value. When the syntax structure is included in another syntax structure, the variable names  $\text{OutSign}$ ,  $\text{OutExp}$ ,  $\text{OutMantissa}$  and  $\text{OutManLen}$  are to be interpreted as being replaced by the variable names used when the syntax structure is included.

**[00513]**  $\text{da\_sign\_flag}$  equal to 0 indicates that the sign of the floating-point value is positive.  $\text{da\_sign\_flag}$  equal to 1 indicates that the sign is negative. The variable  $\text{OutSign}$  is set equal to  $\text{da\_sign\_flag}$ .

[00514] **da\_exponent** specifies the exponent of the floating-point value. The value of **da\_exponent** shall be in the range of 0 to  $2^7 - 2$ , inclusive. The value  $2^7 - 1$  is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value  $2^7 - 1$  as indicating an unspecified value. The variable **OutExp** is set equal to **da\_exponent**.

[00515] **da\_mantissa\_len\_minus1** plus 1 specifies the number of bits in the **da\_mantissa** syntax element. The value of **da\_mantissa\_len\_minus1** shall be in the range of 0 to 31, inclusive. The variable **OutManLen** is set equal to **da\_mantissa\_len\_minus1** + 1.

[00516] **da\_mantissa** specifies the mantissa of the floating-point value. The variable **OutMantissa** is set equal to **da\_mantissa**.

[00517] **Embodiment 19**

[00518] Depth representation information SEI message

[00519] Depth representation information SEI message syntax

| depth_representation_info( payloadSize ) {                                | Descriptor |
|---------------------------------------------------------------------------|------------|
| <b>z_near_flag</b>                                                        | u(1)       |
| <b>z_far_flag</b>                                                         | u(1)       |
| <b>d_min_flag</b>                                                         | u(1)       |
| <b>d_max_flag</b>                                                         | u(1)       |
| <b>depth_representation_type</b>                                          | ue(v)      |
| if( d_min_flag    d_max_flag )                                            |            |
| <b>disparity_ref_view_id</b>                                              | ue(v)      |
| if( z_near_flag )                                                         |            |
| depth_rep_info_element( ZNearSign, ZNearExp, ZNearMantissa, ZNearManLen ) |            |
| if( z_far_flag )                                                          |            |
| depth_rep_info_element( ZFarSign, ZFarExp, ZFarMantissa, ZFarManLen )     |            |
| if( d_min_flag )                                                          |            |
| depth_rep_info_element( DMinSign, DMinExp, DMinMantissa, DMinManLen )     |            |
| if( d_max_flag )                                                          |            |

|                                                                       |       |
|-----------------------------------------------------------------------|-------|
| depth_rep_info_element( DMaxSign, DMaxExp, DMaxMantissa, DMaxManLen ) |       |
| if( depth_representation_type == 3 ) {                                |       |
| <b>depth_nonlinear_representation_num_minus1</b>                      | ue(v) |
| for( i = 1; i <= depth_nonlinear_representation_num_minus1 + 1; i++ ) |       |
| <b>depth_nonlinear_representation_model[ i ]</b>                      |       |
| }                                                                     |       |
| }                                                                     |       |

|                                                                     |                   |
|---------------------------------------------------------------------|-------------------|
| depth_rep_info_element( OutSign, OutExp, OutMantissa, OutManLen ) { | <b>Descriptor</b> |
| <b>da_sign_flag</b>                                                 | u(1)              |
| Add space                                                           | u(7)              |
| <b>da_mantissa_len_minus1</b>                                       | u(5)              |
| <b>da_mantissa</b>                                                  | u(v)              |
| }                                                                   |                   |

[00520] Depth representation information SEI message semantics

[00521] The syntax elements in the depth representation information (**DRI**) SEI message specify various parameters for auxiliary pictures of type AUX\_DEPTH for the purpose of processing decoded primary and auxiliary pictures prior to rendering on a 3D display, such as view synthesis. Specifically, depth or disparity ranges for depth pictures are specified.

[00522] *When a CVS does not contain an SDI SEI message with sdi\_aux\_id[ i ] equal to 2 for at least one value of i, no picture in the CVS shall be associated with a DRI SEI message.*

[00523] *When an AU contains both an SDI SEI message with sdi\_aux\_id[ i ] equal to 2 for at least one value of i and a DRI SEI message, the SDI SEI message shall precede the DRI SEI message in decoding order.*

[00524] When present, the depth representation information SEI message shall be associated with one or more layers *that are indicated as depth auxiliary layers by an SDI SEI message with sdi\_aux\_id value equal to AUX\_DEPTH*. The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the depth representation information SEI message applies.

[00525] When present, the depth representation information SEI message may be included in any access unit. It is recommended that, when present, the SEI message is included for the purpose of random access in an access unit in which the coded picture with `nuh_layer_id` equal to `targetLayerId` is an IRAP picture.

[00526] For an auxiliary picture with `sdi_aux_id[ targetLayerId ]` equal to `AUX_DEPTH`, an associated primary picture, if any, is a picture in the same access unit having `sdi_aux_id[ nuhLayerIdB ]` equal to 0 such that `ScalabilityId[ LayerIdxInVps[ targetLayerId ] ][ j ]` is equal to `ScalabilityId[ LayerIdxInVps[ nuhLayerIdB ] ][ j ]` for all values of `j` in the range of 0 to 2, inclusive, and 4 to 15, inclusive.

[00527] The information indicated in the SEI message applies to all the pictures with `nuh_layer_id` equal to `targetLayerId` from the access unit containing the SEI message up to but excluding the next picture, in decoding order, associated with a depth representation information SEI message applicable to `targetLayerId` or to the end of the CLVS of the `nuh_layer_id` equal to `targetLayerId`, whichever is earlier in decoding order.

[00528] `z_near_flag` equal to 0 specifies that the syntax elements specifying the nearest depth value are not present in the syntax structure. `z_near_flag` equal to 1 specifies that the syntax elements specifying the nearest depth value are present in the syntax structure.

[00529] `z_far_flag` equal to 0 specifies that the syntax elements specifying the farthest depth value are not present in the syntax structure. `z_far_flag` equal to 1 specifies that the syntax elements specifying the farthest depth value are present in the syntax structure.

[00530] `d_min_flag` equal to 0 specifies that the syntax elements specifying the minimum disparity value are not present in the syntax structure. `d_min_flag` equal to 1 specifies that the syntax elements specifying the minimum disparity value are present in the syntax structure.

[00531] `d_max_flag` equal to 0 specifies that the syntax elements specifying the maximum disparity value are not present in the syntax structure. `d_max_flag` equal to 1 specifies that the syntax elements specifying the maximum disparity value are present in the syntax structure.

[00532] `depth_representation_type` specifies the representation definition of decoded luma samples of auxiliary pictures as specified in Table Y1. In Table Y1, disparity specifies the horizontal displacement between two texture views and `Z` value specifies the distance from a camera.

[00533] The variable `maxVal` is set equal to  $(1 \ll (8 + \text{sps\_bitdepth\_minus8})) - 1$ , where `sps_bitdepth_minus8` is the value included in or inferred for the active SPS of the layer with `nuh_layer_id` equal to `targetLayerId`.

Table Y1 – Definition of depth\_representation\_type

| depth_representation_type | Interpretation                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                         | <p>Each decoded luma sample value of an auxiliary picture represents an inverse of Z value that is uniformly quantized into the range of 0 to maxVal, inclusive.</p> <p>When z_far_flag is equal to 1, the luma sample value equal to 0 represents the inverse of ZFar (specified below). When z_near_flag is equal to 1, the luma sample value equal to maxVal represents the inverse of ZNear (specified below).</p>                                                    |
| 1                         | <p>Each decoded luma sample value of an auxiliary picture represents disparity that is uniformly quantized into the range of 0 to maxVal, inclusive.</p> <p>When d_min_flag is equal to 1, the luma sample value equal to 0 represents DMin (specified below). When d_max_flag is equal to 1, the luma sample value equal to maxVal represents DMax (specified below).</p>                                                                                                |
| 2                         | <p>Each decoded luma sample value of an auxiliary picture represents a Z value uniformly quantized into the range of 0 to maxVal, inclusive.</p> <p>When z_far_flag is equal to 1, the luma sample value equal to 0 corresponds to ZFar (specified below). When z_near_flag is equal to 1, the luma sample value equal to maxVal represents ZNear (specified below).</p>                                                                                                  |
| 3                         | <p>Each decoded luma sample value of an auxiliary picture represents a nonlinearly mapped disparity, normalized in range from 0 to maxVal, as specified by depth_nonlinear_representation_num_minus1 and depth_nonlinear_representation_model[ i ].</p> <p>When d_min_flag is equal to 1, the luma sample value equal to 0 represents DMin (specified below). When d_max_flag is equal to 1, the luma sample value equal to maxVal represents DMax (specified below).</p> |

|              |                         |
|--------------|-------------------------|
| Other values | Reserved for future use |
|--------------|-------------------------|

[00534] **disparity\_ref\_view\_id** specifies the ViewId value against which the disparity values are derived.

[00535] NOTE 1 – **disparity\_ref\_view\_id** is present only if **d\_min\_flag** is equal to 1 or **d\_max\_flag** is equal to 1 and is useful for **depth\_representation\_type** values equal to 1 and 3.

[00536] The variables in the x column of Table Y2 are derived from the respective variables in the s, e, n and v columns of Table Y2 as follows:

- If the value of e is in the range of 0 to 127, exclusive, x is set equal to  $(-1)^s * 2^{e-31} * (1 + n \div 2^v)$ .
- Otherwise (e is equal to 0), x is set equal to  $(-1)^s * 2^{-(30+v)} * n$ .

[00537] NOTE 1 – The above specification is similar to that found in IEC 60559:1989.

Table Y2 – Association between depth parameter variables and syntax elements

| x     | s         | e        | n             | v           |
|-------|-----------|----------|---------------|-------------|
| ZNear | ZNearSign | ZNearExp | ZNearMantissa | ZNearManLen |
| ZFar  | ZFarSign  | ZFarExp  | ZFarMantissa  | ZFarManLen  |
| DMax  | DMaxSign  | DMaxExp  | DMaxMantissa  | DMaxManLen  |
| DMin  | DMinSign  | DMinExp  | DMinMantissa  | DMinManLen  |

[00538] The DMin and DMax values, when present, are specified in units of a luma sample width of the coded picture with ViewId equal to ViewId of the auxiliary picture.

[00539] The units for the ZNear and ZFar values, when present, are identical but unspecified.

[00540] **depth\_nonlinear\_representation\_num\_minus1** plus 2 specifies the number of piece-wise linear segments for mapping of depth values to a scale that is uniformly quantized in terms of disparity.

[00541] **depth\_nonlinear\_representation\_model[i]** for i ranging from 0 to **depth\_nonlinear\_representation\_num\_minus1** + 2, inclusive, specify the piece-wise linear segments for mapping of decoded luma sample values of an auxiliary picture to a scale that is uniformly quantized in terms of disparity. The values of **depth\_nonlinear\_representation\_model[0]** and **depth\_nonlinear\_representation\_model[depth\_nonlinear\_representation\_num\_minus1 + 2]** are both inferred to be equal to 0.

**[00542]** NOTE 2 – When `depth_representation_type` is equal to 3, an auxiliary picture contains nonlinearly transformed depth samples. The variable `DepthLUT[ i ]`, as specified below, is used to transform decoded depth sample values from the nonlinear representation to the linear representation, i.e., uniformly quantized disparity values. The shape of this transform is defined by means of line-segment approximation in two-dimensional linear-disparity-to-nonlinear-disparity space. The first  $(0, 0)$  and the last  $(\text{maxVal}, \text{maxVal})$  nodes of the curve are predefined. Positions of additional nodes are transmitted in form of deviations (`depth_nonlinear_representation_model[ i ]`) from the straight-line curve. These deviations are uniformly distributed along the whole range of 0 to `maxVal`, inclusive, with spacing depending on the value of `nonlinear_depth_representation_num_minus1`.

**[00543]** The variable `DepthLUT[ i ]` for  $i$  in the range of 0 to `maxVal`, inclusive, is specified as follows:

```
for( k = 0; k <= depth_nonlinear_representation_num_minus1 + 1; k++ ) {
    pos1 = ( maxVal * k ) / ( depth_nonlinear_representation_num_minus1 + 2 )
    dev1 = depth_nonlinear_representation_model[ k ]
    pos2 = ( maxVal * ( k + 1 ) ) / ( depth_nonlinear_representation_num_minus1 + 2 )
    dev2 = depth_nonlinear_representation_model[ k + 1 ] (X)

    x1 = pos1 - dev1
    y1 = pos1 + dev1
    x2 = pos2 - dev2
    y2 = pos2 + dev2

    for( x = Max( x1, 0 ); x <= Min( x2, maxVal ); x++ )
        DepthLUT[ x ] = Clip3( 0, maxVal, Round( ( ( x - x1 ) * ( y2 - y1 ) ) ÷
        x2 - x1 ) + y1 ) )
}
```

**[00544]** When `depth_representation_type` is equal to 3, `DepthLUT[ dS ]` for all decoded luma sample values `dS` of an auxiliary picture in the range of 0 to `maxVal`, inclusive, represents disparity that is uniformly quantized into the range of 0 to `maxVal`, inclusive.

**[00545]** The syntax structure specifies the value of an element in the depth representation information SEI message.

**[00546]** The syntax structure sets the values of the `OutSign`, `OutExp`, `OutMantissa` and `OutManLen` variables that represent a floating-point value. When the syntax structure is included

in another syntax structure, the variable names OutSign, OutExp, OutMantissa and OutManLen are to be interpreted as being replaced by the variable names used when the syntax structure is included.

[00547] **da\_sign\_flag** equal to 0 indicates that the sign of the floating-point value is positive. da\_sign\_flag equal to 1 indicates that the sign is negative. The variable OutSign is set equal to da\_sign\_flag.

[00548] **da\_exponent** specifies the exponent of the floating-point value. The value of da\_exponent shall be in the range of 0 to  $2^7 - 2$ , inclusive. The value  $2^7 - 1$  is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value  $2^7 - 1$  as indicating an unspecified value. The variable OutExp is set equal to da\_exponent.

[00549] **da\_mantissa\_len\_minus1** plus 1 specifies the number of bits in the da\_mantissa syntax element. The value of da\_mantissa\_len\_minus1 shall be in the range of 0 to 31, inclusive. The variable OutManLen is set equal to da\_mantissa\_len\_minus1 + 1.

[00550] **da\_mantissa** specifies the mantissa of the floating-point value. The variable OutMantissa is set equal to da\_mantissa.

[00551] **Embodiment 20**

[00552] Alpha channel information SEI message

[00553] Alpha channel information SEI message syntax

| alpha_channel_info( payloadSize ) {   | Descriptor |
|---------------------------------------|------------|
| <b>alpha_channel_cancel_flag</b>      | u(1)       |
| if( !alpha_channel_cancel_flag ) {    |            |
| <b>alpha_channel_use_idc</b>          | u(3)       |
| <b>alpha_channel_bit_depth_minus8</b> | u(3)       |
| <b>alpha_transparent_value</b>        | u(v)       |
| <b>alpha_opaque_value</b>             | u(v)       |
| <b>alpha_channel_incr_flag</b>        | u(1)       |
| <b>alpha_channel_clip_flag</b>        | u(1)       |
| if( alpha_channel_clip_flag )         |            |
| <b>alpha_channel_clip_type_flag</b>   | u(1)       |
| }                                     |            |

|   |  |
|---|--|
| } |  |
|---|--|

**[00554]** Alpha channel information SEI message semantics

**[00555]** The alpha channel information SEI message provides information about alpha channel sample values and post-processing applied to the decoded alpha planes coded in auxiliary pictures of type AUX\_ALPHA and one or more associated primary pictures.

**[00556]** For an auxiliary picture with nuh\_layer\_id equal to nuhLayerIdA and sdi\_aux\_id[nuhLayerIdA] equal to AUX\_ALPHA, an associated primary picture, if any, is a picture in the same access unit having sdi\_aux\_id[nuhLayerIdB] equal to 0 such that ScalabilityId[LayerIdxInVps[nuhLayerIdA]][j] is equal to ScalabilityId[LayerIdxInVps[nuhLayerIdB]][j] for all values of j in the range of 0 to 2, inclusive, and 4 to 15, inclusive.

**[00557]** When an access unit contains an auxiliary picture picA with nuh\_layer\_id equal to nuhLayerIdA and sdi\_aux\_id[nuhLayerIdA] equal to AUX\_ALPHA, the alpha channel sample values of picA persist in output order until one or more of the following conditions are true:

- The next picture, in output order, with nuh\_layer\_id equal to nuhLayerIdA is output.
- A CLVS containing the auxiliary picture picA ends.
- The bitstream ends.
- A CLVS of any associated primary layer of the auxiliary picture layer with nuh\_layer\_id equal to nuhLayerIdA ends.

**[00558]** *It is a requirement of bitstream conformance that the alpha channel information SEI message shall not be present in the bitstream in which the scalability dimension information SEI message is not present.*

**[00559]** The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the alpha channel information SEI message applies.

**[00560]** **alpha\_channel\_cancel\_flag** equal to 1 indicates that the alpha channel information SEI message cancels the persistence of any previous alpha channel information SEI message in output order that applies to the current layer. **alpha\_channel\_cancel\_flag** equal to 0 indicates that alpha channel information follows.

**[00561]** Let currPic be the picture that the alpha channel information SEI message is associated with. The semantics of alpha channel information SEI message persist for the current layer in output order until one or more of the following conditions are true:

- A new CLVS of the current layer begins.

- The bitstream ends.
- A picture picB with nuh\_layer\_id equal to targetLayerId in an access unit containing an alpha channel information SEI message with nuh\_layer\_id equal to targetLayerId is output having PicOrderCnt( picB ) greater than PicOrderCnt( currPic ), where PicOrderCnt( picB ) and PicOrderCnt( currPic ) are the PicOrderCntVal values of picB and currPic, respectively, immediately after the invocation of the decoding process for picture order count for picB.

**[00562] alpha\_channel\_use\_idc** equal to 0 indicates that for alpha blending purposes the decoded samples of the associated primary picture should be multiplied by the interpretation sample values of the auxiliary coded picture in the display process after output from the decoding process. alpha\_channel\_use\_idc equal to 1 indicates that for alpha blending purposes the decoded samples of the associated primary picture should not be multiplied by the interpretation sample values of the auxiliary coded picture in the display process after output from the decoding process. alpha\_channel\_use\_idc equal to 2 indicates that the usage of the auxiliary picture is unspecified. Values greater than 2 for alpha\_channel\_use\_idc are reserved for future use by ITU-T | ISO/IEC. When not present, the value of alpha\_channel\_use\_idc is inferred to be equal to 2.

**[00563] alpha\_channel\_bit\_depth\_minus8** plus 8 specifies the bit depth of the samples of the luma sample array of the auxiliary picture. alpha\_channel\_bit\_depth\_minus8 shall be in the range 0 to 7 inclusive. alpha\_channel\_bit\_depth\_minus8 shall be equal to bit\_depth\_luma\_minus8 of the associated primary picture.

**[00564] alpha\_transparent\_value** specifies the interpretation sample value of an auxiliary coded picture luma sample for which the associated luma and chroma samples of the primary coded picture are considered transparent for purposes of alpha blending. The number of bits used for the representation of the alpha\_transparent\_value syntax element is alpha\_channel\_bit\_depth\_minus8 + 9.

**[00565] alpha\_opaque\_value** specifies the interpretation sample value of an auxiliary coded picture luma sample for which the associated luma and chroma samples of the primary coded picture are considered opaque for purposes of alpha blending. The number of bits used for the representation of the alpha\_opaque\_value syntax element is alpha\_channel\_bit\_depth\_minus8 + 9.

**[00566] alpha\_channel\_incr\_flag** equal to 0 indicates that the interpretation sample value for each decoded auxiliary picture luma sample value is equal to the decoded auxiliary picture sample value for purposes of alpha blending. alpha\_channel\_incr\_flag equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample value that is greater than Min( alpha\_opaque\_value, alpha\_transparent\_value )

should be increased by one to obtain the interpretation sample value for the auxiliary picture sample and any auxiliary picture luma sample value that is less than or equal to  $\text{Min}(\text{alpha\_opaque\_value}, \text{alpha\_transparent\_value})$  should be used, without alteration, as the interpretation sample value for the decoded auxiliary picture sample value. When not present, the value of `alpha_channel_incr_flag` is inferred to be equal to 0.

[00567] **alpha\_channel\_clip\_flag** equal to 0 indicates that no clipping operation is applied to obtain the interpretation sample values of the decoded auxiliary picture. `alpha_channel_clip_flag` equal to 1 indicates that the interpretation sample values of the decoded auxiliary picture are altered according to the clipping process described by the `alpha_channel_clip_type_flag` syntax element. When not present, the value of `alpha_channel_clip_flag` is inferred to be equal to 0.

[00568] **alpha\_channel\_clip\_type\_flag** equal to 0 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample that is greater than  $(\text{alpha\_opaque\_value} - \text{alpha\_transparent\_value}) / 2$  is set equal to `alpha_opaque_value` to obtain the interpretation sample value for the auxiliary picture luma sample and any auxiliary picture luma sample that is less or equal than  $(\text{alpha\_opaque\_value} - \text{alpha\_transparent\_value}) / 2$  is set equal to `alpha_transparent_value` to obtain the interpretation sample value for the auxiliary picture luma sample. `alpha_channel_clip_type_flag` equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample that is greater than `alpha_opaque_value` is set equal to `alpha_opaque_value` to obtain the interpretation sample value for the auxiliary picture luma sample and any auxiliary picture luma sample that is less than or equal to `alpha_transparent_value` is set equal to `alpha_transparent_value` to obtain the interpretation sample value for the auxiliary picture luma sample.

[00569] NOTE – When both `alpha_channel_incr_flag` and `alpha_channel_clip_flag` are equal to one, the clipping operation specified by `alpha_channel_clip_type_flag` should be applied first followed by the alteration specified by `alpha_channel_incr_flag` to obtain the interpretation sample value for the auxiliary picture luma sample.

[00570] **Embodiment 21**

[00571] Alpha channel information SEI message

[00572] Alpha channel information SEI message syntax

|                                                  |                   |
|--------------------------------------------------|-------------------|
| <code>alpha_channel_info( payloadSize ) {</code> | <b>Descriptor</b> |
| <b><code>alpha_channel_cancel_flag</code></b>    | <code>u(1)</code> |
| <code>if( !alpha_channel_cancel_flag ) {</code>  |                   |

|                                       |      |
|---------------------------------------|------|
| <b>alpha_channel_use_idc</b>          | u(3) |
| <b>alpha_channel_bit_depth_minus8</b> | u(3) |
| <b>alpha_transparent_value</b>        | u(v) |
| <b>alpha_opaque_value</b>             | u(v) |
| <b>alpha_channel_incr_flag</b>        | u(1) |
| <b>alpha_channel_clip_flag</b>        | u(1) |
| if( alpha_channel_clip_flag )         |      |
| <b>alpha_channel_clip_type_flag</b>   | u(1) |
| }                                     |      |
| }                                     |      |

[00573] Alpha channel information SEI message semantics

[00574] The alpha channel information (**ACI**) SEI message provides information about alpha channel sample values and post-processing applied to the decoded alpha planes coded in auxiliary pictures of type AUX\_ALPHA and one or more associated primary pictures.

[00575] *For an auxiliary picture with nuh\_layer\_id equal to nuhLayerIdA and sdi\_aux\_id[nuhLayerIdA] equal to AUX\_ALPHA, an associated primary picture, if any, is a picture in the same access unit having sdi\_aux\_id[nuhLayerIdB] equal to 0 such that ScalabilityId[ LayerIdxInVps[ nuhLayerIdA ] ][j] is equal to ScalabilityId[ LayerIdxInVps[ nuhLayerIdB ] ][j] for all values of j in the range of 0 to 2, inclusive, and 4 to 15, inclusive.*

[00576] *When a CVS does not contain an SDI SEI message with sdi\_aux\_id[i] equal to 1 for at least one value of i, no picture in the CVS shall be associated with an ACI SEI message.*

[00577] *When an AU contains both an SDI SEI message with sdi\_aux\_id[i] equal to 1 for at least one value of i and an ACI SEI message, the SDI SEI message shall precede the ACI SEI message in decoding order.*

[00578] When an access unit contains an auxiliary picture picA **in a layer that is indicated as an alpha auxiliary layer by an SDI SEI message with nuh\_layer\_id equal to nuhLayerIdA and sdi\_aux\_id[nuhLayerIdA] equal to AUX\_ALPHA**, the alpha channel sample values of picA persist in output order until one or more of the following conditions are true:

[00579] The next picture, in output order, with nuh\_layer\_id equal to nuhLayerIdA is output.

- A CLVS containing the auxiliary picture picA ends.

- The bitstream ends.
- A CLVS of any associated primary layer of the auxiliary picture layer with nuh\_layer\_id equal to nuhLayerIdA ends.

[00580] The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the alpha channel information SEI message applies.

[00581] **alpha\_channel\_cancel\_flag** equal to 1 indicates that the alpha channel information SEI message cancels the persistence of any previous alpha channel information SEI message in output order that applies to the current layer. alpha\_channel\_cancel\_flag equal to 0 indicates that alpha channel information follows.

[00582] Let currPic be the picture that the alpha channel information SEI message is associated with. The semantics of alpha channel information SEI message persist for the current layer in output order until one or more of the following conditions are true:

- A new CLVS of the current layer begins.
- The bitstream ends.
- A picture *in the current layer in an AU associated with picB with nuh\_layer\_id equal to targetLayerId in an access unit containing an alpha channel information SEI message is output that follows the current picture in output order. with nuh\_layer\_id equal to targetLayerId is output having PicOrderCnt(picB) greater than PicOrderCnt(currPic), where PicOrderCnt(picB) and PicOrderCnt(currPic) are the PicOrderCntVal values of picB and currPic, respectively, immediately after the invocation of the decoding process for picture order count for picB.*

[00583] **alpha\_channel\_use\_idc** equal to 0 indicates that for alpha blending purposes the decoded samples of the associated primary picture should be multiplied by the interpretation sample values of the auxiliary coded picture in the display process after output from the decoding process. alpha\_channel\_use\_idc equal to 1 indicates that for alpha blending purposes the decoded samples of the associated primary picture should not be multiplied by the interpretation sample values of the auxiliary coded picture in the display process after output from the decoding process. alpha\_channel\_use\_idc equal to 2 indicates that the usage of the auxiliary picture is unspecified. Values greater than 2 for alpha\_channel\_use\_idc are reserved for future use by ITU-T | ISO/IEC. When not present, the value of alpha\_channel\_use\_idc is inferred to be equal to 2.

[00584] **alpha\_channel\_bit\_depth\_minus8** plus 8 specifies the bit depth of the samples of the luma sample array of the auxiliary picture. alpha\_channel\_bit\_depth\_minus8 shall be in the range 0 to 7 inclusive. alpha\_channel\_bit\_depth\_minus8 shall be equal to bit\_depth\_luma\_minus8 of the associated primary picture.

[00585] **alpha\_transparent\_value** specifies the interpretation sample value of an auxiliary coded picture luma sample for which the associated luma and chroma samples of the primary coded picture are considered transparent for purposes of alpha blending. The number of bits used for the representation of the **alpha\_transparent\_value** syntax element is  $\text{alpha\_channel\_bit\_depth\_minus8} + 9$ .

[00586] **alpha\_opaque\_value** specifies the interpretation sample value of an auxiliary coded picture luma sample for which the associated luma and chroma samples of the primary coded picture are considered opaque for purposes of alpha blending. The number of bits used for the representation of the **alpha\_opaque\_value** syntax element is  $\text{alpha\_channel\_bit\_depth\_minus8} + 9$ .

[00587] **alpha\_channel\_incr\_flag** equal to 0 indicates that the interpretation sample value for each decoded auxiliary picture luma sample value is equal to the decoded auxiliary picture sample value for purposes of alpha blending. **alpha\_channel\_incr\_flag** equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample value that is greater than  $\text{Min}(\text{alpha\_opaque\_value}, \text{alpha\_transparent\_value})$  should be increased by one to obtain the interpretation sample value for the auxiliary picture sample and any auxiliary picture luma sample value that is less than or equal to  $\text{Min}(\text{alpha\_opaque\_value}, \text{alpha\_transparent\_value})$  should be used, without alteration, as the interpretation sample value for the decoded auxiliary picture sample value. When not present, the value of **alpha\_channel\_incr\_flag** is inferred to be equal to 0.

[00588] **alpha\_channel\_clip\_flag** equal to 0 indicates that no clipping operation is applied to obtain the interpretation sample values of the decoded auxiliary picture. **alpha\_channel\_clip\_flag** equal to 1 indicates that the interpretation sample values of the decoded auxiliary picture are altered according to the clipping process described by the **alpha\_channel\_clip\_type\_flag** syntax element. When not present, the value of **alpha\_channel\_clip\_flag** is inferred to be equal to 0.

[00589] **alpha\_channel\_clip\_type\_flag** equal to 0 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample that is greater than  $(\text{alpha\_opaque\_value} - \text{alpha\_transparent\_value}) / 2$  is set equal to **alpha\_opaque\_value** to obtain the interpretation sample value for the auxiliary picture luma sample and any auxiliary picture luma sample that is less or equal than  $(\text{alpha\_opaque\_value} - \text{alpha\_transparent\_value}) / 2$  is set equal to **alpha\_transparent\_value** to obtain the interpretation sample value for the auxiliary picture luma sample. **alpha\_channel\_clip\_type\_flag** equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary picture samples, any auxiliary picture luma sample that is greater than

alpha\_opaque\_value is set equal to alpha\_opaque\_value to obtain the interpretation sample value for the auxiliary picture luma sample and any auxiliary picture luma sample that is less than or equal to alpha\_transparent\_value is set equal to alpha\_transparent\_value to obtain the interpretation sample value for the auxiliary picture luma sample.

[00590] NOTE – When both alpha\_channel\_incr\_flag and alpha\_channel\_clip\_flag are equal to one, the clipping operation specified by alpha\_channel\_clip\_type\_flag should be applied first followed by the alteration specified by alpha\_channel\_incr\_flag to obtain the interpretation sample value for the auxiliary picture luma sample.

[00591] **Embodiment 22**

[00592] Scalability dimension information (SDI) SEI message

[00593] Scalability dimension SEI message syntax

| scalability_dimension( payloadSize ) {                     | Descriptor |
|------------------------------------------------------------|------------|
| <b>sdi_max_layers_minus1</b>                               | u(6)       |
| <b>sdi_multiview_info_flag</b>                             | u(1)       |
| <b>sdi_auxiliary_info_flag</b>                             | u(1)       |
| if( sdi_multiview_info_flag    sdi_auxiliary_info_flag ) { |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_len</b>                                     | u(4)       |
| for( i = 0; i <= sdi_max_layers_minus1; i++ ) {            |            |
| if( sdi_multiview_info_flag )                              |            |
| <b>sdi_view_id_val[ i ]</b>                                | u(v)       |
| if( sdi_auxiliary_info_flag )                              |            |
| <b>sdi_aux_id[ i ]</b>                                     | u(8)       |
| }                                                          |            |
| }                                                          |            |
| }                                                          |            |

[00594] Scalability dimension SEI message semantics

[00595] The scalability dimension SEI message provides the scalability dimension information for each layer in bitstreamInScope (defined below), such as 1) when bitstreamInScope may be a multiview bitstream, the view ID of each layer; and 2) when there

may be auxiliary information (such as depth or alpha) carried by one or more layers in bitstreamInScope, the auxiliary ID of each layer.

[00596] The bitstreamInScope is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension SEI message, followed by zero or more AUs, including all subsequent AUs up to but not including any subsequent AU that contains a scalability dimension SEI message.

[00597] **sdi\_max\_layers\_minus1** plus 1 indicates the maximum number of layers in bitstreamInScope.

[00598] **sdi\_multiview\_info\_flag** equal to 1 indicates that bitstreamInScope may be a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_multiview\_flag** equal to 0 indicates that bitstreamInScope is not a multiview bitstream and the **sdi\_view\_id\_val[ ]** syntax elements are not present in the scalability dimension SEI message.

[00599] **sdi\_auxiliary\_info\_flag** equal to 1 indicates that there may be auxiliary information carried by one or more layers in bitstreamInScope and the **sdi\_aux\_id[ ]** syntax elements are present in the scalability dimension SEI message. **sdi\_auxiliary\_info\_flag** equal to 0 indicates that there is no auxiliary information carried by one or more layers in bitstreamInScope and the **sdi\_aux\_id[ ]** syntax elements are not present in the scalability dimension SEI message.

[00600] **sdi\_view\_id\_len** specifies the length, in bits, of the **sdi\_view\_id\_val[ i ]** syntax element.

[00601] **sdi\_view\_id\_val[ i ]** specifies the view ID of the i-th layer in bitstreamInScope. The length of the **sdi\_view\_id\_val[ i ]** syntax element is **sdi\_view\_id\_len** bits. When not present, the value of **sdi\_view\_id\_val[ i ]** is inferred to be equal to 0.

[00602] *The variable NumViews is derived as follows:*

```

NumViews = 1
if ( sdi_multiview_info_flag ) {
    for ( i = 1; i <= sdi_max_layers_minus1; i++ ) {
        newViewFlag = 1
        for ( j = 0; j < i; j++ )           (X)
            if ( sdi_view_id_val[ i ] == sdi_view_id_val[ j ] )
                newViewFlag = 0
        if ( newViewFlag )
            NumViews++
    }
}

```

}

[00603] **sdi\_aux\_id[ i ]** equal to 0 indicates that the i-th layer in bitstreamInScope does not contain auxiliary pictures. **sdi\_aux\_id[ i ]** greater than 0 indicates the type of auxiliary pictures in the i-th layer in bitstreamInScope as specified in Table 1.

Table 1 – Mapping of **sdi\_aux\_id[ i ]** to the type of auxiliary pictures

| <b>sdi_aux_id[ i ]</b> | <b>Name</b> | <b>Type of auxiliary pictures</b> |
|------------------------|-------------|-----------------------------------|
| 1                      | AUX_ALPHA   | Alpha plane                       |
| 2                      | AUX_DEPTH   | Depth picture                     |
| 3..127                 |             | Reserved                          |
| 128..159               |             | Unspecified                       |
| 160..255               |             | Reserved                          |

[00604] NOTE 3 – The interpretation of auxiliary pictures associated with **sdi\_aux\_id** in the range of 128 to 159, inclusive, is specified through means other than the **sdi\_aux\_id** value.

[00605] **sdi\_aux\_id[ i ]** shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this Specification. Although the value of **sdi\_aux\_id[ i ]** shall be in the range of 0 to 2, inclusive, or 128 to 159, inclusive, in this version of this Specification, decoders shall allow values of **sdi\_aux\_id[ i ]** in the range of 0 to 255, inclusive.

[00606] Multiview acquisition information SEI message

[00607] Multiview acquisition information SEI message syntax

| <b>multiview_acquisition_info( payloadSize ) {</b>                                      | <b>Descriptor</b> |
|-----------------------------------------------------------------------------------------|-------------------|
| <b>intrinsic_param_flag</b>                                                             | u(1)              |
| <b>extrinsic_param_flag</b>                                                             | u(1)              |
| if( <b>intrinsic_param_flag</b> ) {                                                     |                   |
| <b>intrinsic_params_equal_flag</b>                                                      | u(1)              |
| <b>prec_focal_length</b>                                                                | ue(v)             |
| <b>prec_principal_point</b>                                                             | ue(v)             |
| <b>prec_skew_factor</b>                                                                 | ue(v)             |
| for( i = 0; i <= <b>intrinsic_params_equal_flag</b> ? 0 : (nNumViews – Minus1); i++ ) { |                   |

|                                         |       |
|-----------------------------------------|-------|
| <b>sign_focal_length_x[ i ]</b>         | u(1)  |
| <b>exponent_focal_length_x[ i ]</b>     | u(6)  |
| <b>mantissa_focal_length_x[ i ]</b>     | u(v)  |
| <b>sign_focal_length_y[ i ]</b>         | u(1)  |
| <b>exponent_focal_length_y[ i ]</b>     | u(6)  |
| <b>mantissa_focal_length_y[ i ]</b>     | u(v)  |
| <b>sign_principal_point_x[ i ]</b>      | u(1)  |
| <b>exponent_principal_point_x[ i ]</b>  | u(6)  |
| <b>mantissa_principal_point_x[ i ]</b>  | u(v)  |
| <b>sign_principal_point_y[ i ]</b>      | u(1)  |
| <b>exponent_principal_point_y[ i ]</b>  | u(6)  |
| <b>mantissa_principal_point_y[ i ]</b>  | u(v)  |
| <b>sign_skew_factor[ i ]</b>            | u(1)  |
| <b>exponent_skew_factor[ i ]</b>        | u(6)  |
| <b>mantissa_skew_factor[ i ]</b>        | u(v)  |
| }                                       |       |
| }                                       |       |
| if( extrinsic_param_flag ) {            |       |
| <b>prec_rotation_param</b>              | ue(v) |
| <b>prec_translation_param</b>           | ue(v) |
| for( i = 0; i <= numViewsMinus1; i++ )  |       |
| for( j = 0; j < 3; j++ ) { /* row */    |       |
| for( k = 0; k < 3; k++ ) { /* column */ |       |
| <b>sign_r[ i ][ j ][ k ]</b>            | u(1)  |
| <b>exponent_r[ i ][ j ][ k ]</b>        | u(6)  |
| <b>mantissa_r[ i ][ j ][ k ]</b>        | u(v)  |
| }                                       |       |
| <b>sign_t[ i ][ j ]</b>                 | u(1)  |

|                             |      |
|-----------------------------|------|
| <b>exponent_t[ i ][ j ]</b> | u(6) |
| <b>mantissa_t[ i ][ j ]</b> | u(v) |
| }                           |      |
| }                           |      |
| }                           |      |

[00608] Multiview acquisition information SEI message semantics

[00609] The multiview acquisition information (*MAI*) SEI message specifies various parameters of the acquisition environment. Specifically, intrinsic and extrinsic camera parameters are specified. These parameters could be used for processing the decoded views prior to rendering on a 3D display.

[00610] *An MAI SEI message that has payloadType equal to 179 (multiview acquisition) shall not be contained in a scalable nesting SEI message.*

[00611] *Let the current AU be the AU containing the current MAI SEI message, and the current CVS be the CVS containing the current AU.*

[00612] *When a CVS does not contain an SDI SEI message, the CVS shall not contain an MAI SEI message.*

[00613] *When an AU contains both an SDI SEI message and an MAI SEI message, the SDI SEI message shall precede the MAI SEI message in decoding order.*

[00614] The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the multiview acquisition information SEI message applies.

[00615] When present, the multiview acquisition information SEI message that applies to the current layer shall be included in an access unit that contains an IRAP picture that is the first picture of a CLVS of the current layer. The information signalled in the SEI message applies to the CLVS.

[00616] *When the multiview acquisition information SEI message is contained in a scalable nesting SEI message, the syntax elements sn\_ols\_flag and sn\_all\_layers\_flag in the scalable nesting SEI message shall be equal to 0.*

[00617] *The variable numViewsMinus1 is derived as follows:*

- *If the multiview acquisition information SEI message is not included in a scalable nesting SEI message, numViewsMinus1 is set equal to 0.*
- *Otherwise (the multiview acquisition information SEI message is included in a scalable nesting SEI message), numViewsMinus1 is set equal to sn\_num\_layers\_minus1.*

[00618] Some of the views for which the multiview acquisition information is included in a multiview acquisition information SEI message may not be present.

[00619] In the semantics below, index  $i$  refers to the syntax elements and variables that apply to the layer with `nuh_layer_id` equal to `NestingLayerId[  $i$  ]`.

[00620] The extrinsic camera parameters are specified according to a right-handed coordinate system, where the upper left corner of the image is the origin, i.e., the  $(0, 0)$  coordinate, with the other corners of the image having non-negative coordinates. With these specifications, a 3-dimensional world point,  $wP = [x\ y\ z]$  is mapped to a 2-dimensional camera point,  $cP[i] = [u\ v\ 1]$ , for the  $i$ -th camera according to:

$$s * cP[i] = A[i] * R^{-1}[i] * (wP - T[i]) \quad (X)$$

where  $A[i]$  denotes the intrinsic camera parameter matrix,  $R^{-1}[i]$  denotes the inverse of the rotation matrix  $R[i]$ ,  $T[i]$  denotes the translation vector and  $s$  (a scalar value) is an arbitrary scale factor chosen to make the third coordinate of  $cP[i]$  equal to 1. The elements of  $A[i]$ ,  $R[i]$  and  $T[i]$  are determined according to the syntax elements signalled in this SEI message and as specified below.

[00621] **intrinsic\_param\_flag** equal to 1 indicates the presence of intrinsic camera parameters. **intrinsic\_param\_flag** equal to 0 indicates the absence of intrinsic camera parameters.

[00622] **extrinsic\_param\_flag** equal to 1 indicates the presence of extrinsic camera parameters. **extrinsic\_param\_flag** equal to 0 indicates the absence of extrinsic camera parameters.

[00623] **intrinsic\_params\_equal\_flag** equal to 1 indicates that the intrinsic camera parameters are equal for all cameras and only one set of intrinsic camera parameters are present. **intrinsic\_params\_equal\_flag** equal to 0 indicates that the intrinsic camera parameters are different for each camera and that a set of intrinsic camera parameters are present for each camera.

[00624] **prec\_focal\_length** specifies the exponent of the maximum allowable truncation error for `focal_length_x[i]` and `focal_length_y[i]` as given by  $2^{-\text{prec\_focal\_length}}$ . The value of **prec\_focal\_length** shall be in the range of 0 to 31, inclusive.

[00625] **prec\_principal\_point** specifies the exponent of the maximum allowable truncation error for `principal_point_x[i]` and `principal_point_y[i]` as given by  $2^{-\text{prec\_principal\_point}}$ . The value of **prec\_principal\_point** shall be in the range of 0 to 31, inclusive.

[00626] **prec\_skew\_factor** specifies the exponent of the maximum allowable truncation error for skew factor as given by  $2^{-\text{prec\_skew\_factor}}$ . The value of **prec\_skew\_factor** shall be in the range of 0 to 31, inclusive.

[00627] **sign\_focal\_length\_x[ i ]** equal to 0 indicates that the sign of the focal length of the i-th camera in the horizontal direction is positive. **sign\_focal\_length\_x[ i ]** equal to 1 indicates that the sign is negative.

[00628] **exponent\_focal\_length\_x[ i ]** specifies the exponent part of the focal length of the i-th camera in the horizontal direction. The value of **exponent\_focal\_length\_x[ i ]** shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified focal length.

[00629] **mantissa\_focal\_length\_x[ i ]** specifies the mantissa part of the focal length of the i-th camera in the horizontal direction. The length of the **mantissa\_focal\_length\_x[ i ]** syntax element is variable and determined as follows:

- If **exponent\_focal\_length\_x[ i ]** is equal to 0, the length is  $\text{Max}(0, \text{prec\_focal\_length} - 30)$ .
- Otherwise (**exponent\_focal\_length\_x[ i ]** is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_focal\_length\_x}[ i ] + \text{prec\_focal\_length} - 31)$ .

[00630] **sign\_focal\_length\_y[ i ]** equal to 0 indicates that the sign of the focal length of the i-th camera in the vertical direction is positive. **sign\_focal\_length\_y[ i ]** equal to 1 indicates that the sign is negative.

[00631] **exponent\_focal\_length\_y[ i ]** specifies the exponent part of the focal length of the i-th camera in the vertical direction. The value of **exponent\_focal\_length\_y[ i ]** shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified focal length.

[00632] **mantissa\_focal\_length\_y[ i ]** specifies the mantissa part of the focal length of the i-th camera in the vertical direction.

[00633] The length of the **mantissa\_focal\_length\_y[ i ]** syntax element is variable and determined as follows:

- If **exponent\_focal\_length\_y[ i ]** is equal to 0, the length is  $\text{Max}(0, \text{prec\_focal\_length} - 30)$ .
- Otherwise (**exponent\_focal\_length\_y[ i ]** is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_focal\_length\_y}[ i ] + \text{prec\_focal\_length} - 31)$ .

[00634] **sign\_principal\_point\_x[ i ]** equal to 0 indicates that the sign of the principal point of the i-th camera in the horizontal direction is positive. **sign\_principal\_point\_x[ i ]** equal to 1 indicates that the sign is negative.

[00635] **exponent\_principal\_point\_x[ i ]** specifies the exponent part of the principal point of the i-th camera in the horizontal direction. The value of **exponent\_principal\_point\_x[ i ]** shall be

in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified principal point.

**[00636] mantissa\_principal\_point\_x[ i ]** specifies the mantissa part of the principal point of the i-th camera in the horizontal direction. The length of the mantissa\_principal\_point\_x[ i ] syntax element in units of bits is variable and is determined as follows:

- If exponent\_principal\_point\_x[ i ] is equal to 0, the length is  $\text{Max}(0, \text{prec\_principal\_point} - 30)$ .
- Otherwise (exponent\_principal\_point\_x[ i ] is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_principal\_point\_x}[ i ] + \text{prec\_principal\_point} - 31)$ .

**[00637] sign\_principal\_point\_y[ i ]** equal to 0 indicates that the sign of the principal point of the i-th camera in the vertical direction is positive. sign\_principal\_point\_y[ i ] equal to 1 indicates that the sign is negative.

**[00638] exponent\_principal\_point\_y[ i ]** specifies the exponent part of the principal point of the i-th camera in the vertical direction. The value of exponent\_principal\_point\_y[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified principal point.

**[00639] mantissa\_principal\_point\_y[ i ]** specifies the mantissa part of the principal point of the i-th camera in the vertical direction. The length of the mantissa\_principal\_point\_y[ i ] syntax element in units of bits is variable and is determined as follows:

- If exponent\_principal\_point\_y[ i ] is equal to 0, the length is  $\text{Max}(0, \text{prec\_principal\_point} - 30)$ .
- Otherwise (exponent\_principal\_point\_y[ i ] is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_principal\_point\_y}[ i ] + \text{prec\_principal\_point} - 31)$ .

**[00640] sign\_skew\_factor[ i ]** equal to 0 indicates that the sign of the skew factor of the i-th camera is positive.

**[00641] sign\_skew\_factor[ i ]** equal to 1 indicates that the sign is negative.

**[00642] exponent\_skew\_factor[ i ]** specifies the exponent part of the skew factor of the i-th camera. The value of exponent\_skew\_factor[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified skew factor.

**[00643] mantissa\_skew\_factor[ i ]** specifies the mantissa part of the skew factor of the i-th camera. The length of the mantissa\_skew\_factor[ i ] syntax element is variable and determined as follows:

- If  $\text{exponent\_skew\_factor}[i]$  is equal to 0, the length is  $\text{Max}(0, \text{prec\_skew\_factor} - 30)$ .
- Otherwise ( $\text{exponent\_skew\_factor}[i]$  is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_skew\_factor}[i] + \text{prec\_skew\_factor} - 31)$ .

**[00644]** The intrinsic matrix  $A[i]$  for  $i$ -th camera is represented by:

$$\begin{bmatrix} \text{focalLengthX}[i] & \text{skewFactor}[i] & \text{principalPointX}[i] \\ 0 & \text{focalLengthY}[i] & \text{principalPointY}[i] \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{X})$$

**[00645]**  $\text{prec\_rotation\_param}$  specifies the exponent of the maximum allowable truncation error for  $r[i][j][k]$  as given by  $2^{-\text{prec\_rotation\_param}}$ . The value of  $\text{prec\_rotation\_param}$  shall be in the range of 0 to 31, inclusive.

**[00646]**  $\text{prec\_translation\_param}$  specifies the exponent of the maximum allowable truncation error for  $t[i][j]$  as given by  $2^{-\text{prec\_translation\_param}}$ . The value of  $\text{prec\_translation\_param}$  shall be in the range of 0 to 31, inclusive.

**[00647]**  $\text{sign\_r}[i][j][k]$  equal to 0 indicates that the sign of  $(j, k)$  component of the rotation matrix for the  $i$ -th camera is positive.  $\text{sign\_r}[i][j][k]$  equal to 1 indicates that the sign is negative.

**[00648]**  $\text{exponent\_r}[i][j][k]$  specifies the exponent part of  $(j, k)$  component of the rotation matrix for the  $i$ -th camera. The value of  $\text{exponent\_r}[i][j][k]$  shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified rotation matrix.

**[00649]**  $\text{mantissa\_r}[i][j][k]$  specifies the mantissa part of  $(j, k)$  component of the rotation matrix for the  $i$ -th camera. The length of the  $\text{mantissa\_r}[i][j][k]$  syntax element in units of bits is variable and determined as follows:

- If  $\text{exponent\_r}[i]$  is equal to 0, the length is  $\text{Max}(0, \text{prec\_rotation\_param} - 30)$ .
- Otherwise ( $\text{exponent\_r}[i]$  is in the range of 0 to 63, exclusive), the length is  $\text{Max}(0, \text{exponent\_r}[i] + \text{prec\_rotation\_param} - 31)$ .

**[00650]** The rotation matrix  $R[i]$  for  $i$ -th camera is represented as follows:

$$\begin{bmatrix} rE[i][0][0] & rE[i][0][1] & rE[i][0][2] \\ rE[i][1][0] & rE[i][1][1] & rE[i][1][2] \\ rE[i][2][0] & rE[i][2][1] & rE[i][2][2] \end{bmatrix} \quad (\text{X})$$

**[00651]**  $\text{sign\_t}[i][j]$  equal to 0 indicates that the sign of the  $j$ -th component of the translation vector for the  $i$ -th camera is positive.  $\text{sign\_t}[i][j]$  equal to 1 indicates that the sign is negative.

[00652] **exponent\_t**[ i ][ j ] specifies the exponent part of the j-th component of the translation vector for the i-th camera. The value of **exponent\_t**[ i ][ j ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified translation vector.

[00653] **mantissa\_t**[ i ][ j ] specifies the mantissa part of the j-th component of the translation vector for the i-th camera. The length *v* of the **mantissa\_t**[ i ][ j ] syntax element in units of bits is variable and is determined as follows:

- If **exponent\_t**[ i ] is equal to 0, the length *v* is set equal to  $\text{Max}(0, \text{prec\_translation\_param} - 30)$ .
- Otherwise ( $0 < \text{exponent\_t}[ i ] < 63$ ), the length *v* is set equal to  $\text{Max}(0, \text{exponent\_t}[ i ] + \text{prec\_translation\_param} - 31)$ .

[00654] The translation vector **T**[ i ] for the i-th camera is represented by:

$$\begin{bmatrix} \text{tE}[i][0] \\ \text{tE}[i][1] \\ \text{tE}[i][2] \end{bmatrix} \quad (\text{X})$$

[00655] The association between the camera parameter variables and corresponding syntax elements is specified by Table ZZ. Each component of the intrinsic and rotation matrices and the translation vector is obtained from the variables specified in Table ZZ as the variable *x* computed as follows:

- If *e* is in the range of 0 to 63, exclusive, *x* is set equal to  $(-1)^s * 2^{e-31} * (1 + n \div 2^v)$ .
- Otherwise (*e* is equal to 0), *x* is set equal to  $(-1)^s * 2^{-(30+v)} * n$ .

[00656] NOTE – The above specification is similar to that found in IEC 60559:1989.

Table ZZ – Association between camera parameter variables and syntax elements.

| <b>x</b>                     | <b>s</b>                    | <b>e</b>                        | <b>n</b>                        |
|------------------------------|-----------------------------|---------------------------------|---------------------------------|
| <b>focalLengthX</b> [ i ]    | sign_focal_length_x[ i ]    | exponent_focal_length_x[ i ]    | mantissa_focal_length_x[ i ]    |
| <b>focalLengthY</b> [ i ]    | sign_focal_length_y[ i ]    | exponent_focal_length_y[ i ]    | mantissa_focal_length_y[ i ]    |
| <b>principalPointX</b> [ i ] | sign_principal_point_x[ i ] | exponent_principal_point_x[ i ] | mantissa_principal_point_x[ i ] |
| <b>principalPointY</b> [ i ] | sign_principal_point_y[ i ] | exponent_principal_point_y[ i ] | mantissa_principal_point_y[ i ] |

|                           |                          |                              |                              |
|---------------------------|--------------------------|------------------------------|------------------------------|
| <b>skewFactor</b> [ i ]   | sign_skew_factor[ i<br>] | exponent_skew_factor[ i<br>] | mantissa_skew_factor[ i<br>] |
| <b>rE</b> [ i ][ j ][ k ] | sign_r[ i ][ j ][ k ]    | exponent_r[ i ][ j ][ k ]    | mantissa_r[ i ][ j ][ k ]    |
| <b>tE</b> [ i ][ j ]      | sign_t[ i ][ j ]         | exponent_t[ i ][ j ]         | mantissa_t[ i ][ j ]         |

**[00657]** FIG. 4 is a block diagram showing an example video processing system 400 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the video processing system 400. The video processing system 400 may include input 402 for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input 402 may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as Wi-Fi or cellular interfaces.

**[00658]** The video processing system 400 may include a coding component 404 that may implement the various coding or encoding methods described in the present document. The coding component 404 may reduce the average bitrate of video from the input 402 to the output of the coding component 404 to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component 404 may be either stored, or transmitted via a communication connected, as represented by the component 406. The stored or communicated bitstream (or coded) representation of the video received at the input 402 may be used by the component 408 for generating pixel values or displayable video that is sent to a display interface 410. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as “coding” operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

**[00659]** Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on. Examples of storage interfaces include SATA (serial advanced technology attachment), Peripheral Component Interconnect (PCI), Integrated Drive Electronics (IDE) interface, and the like. The techniques described in the present document may be embodied in various electronic

devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

**[00660]** FIG. 5 is a block diagram of a video processing apparatus 500. The apparatus 500 may be used to implement one or more of the methods described herein. The apparatus 500 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 500 may include one or more processors 502, one or more memories 504 and video processing hardware 506 (a.k.a., video processing circuitry). The processor(s) 502 may be configured to implement one or more methods described in the present document. The memory (memories) 504 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 506 may be used to implement, in hardware circuitry, some techniques described in the present document. In some embodiments, the hardware 506 may be partly or completely located within the processor 502, e.g., a graphics processor.

**[00661]** FIG. 6 is a block diagram that illustrates an example video coding system 600 that may utilize the techniques of this disclosure. As shown in FIG. 6, the video coding system 600 may include a source device 610 and a destination device 620. Source device 610 generates encoded video data which may be referred to as a video encoding device. Destination device 620 may decode the encoded video data generated by source device 610 which may be referred to as a video decoding device.

**[00662]** Source device 610 may include a video source 612, a video encoder 614, and an input/output (I/O) interface 616.

**[00663]** Video source 612 may include a source such as a video capture device, an interface to receive video data from a video content provider, and/or a computer graphics system for generating video data, or a combination of such sources. The video data may comprise one or more pictures. Video encoder 614 encodes the video data from video source 612 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. I/O interface 616 may include a modulator/demodulator (modem) and/or a transmitter. The encoded video data may be transmitted directly to destination device 620 via I/O interface 616 through network 630. The encoded video data may also be stored onto a storage medium/server 640 for access by destination device 620.

[00664] Destination device 620 may include an I/O interface 626, a video decoder 624, and a display device 622.

[00665] I/O interface 626 may include a receiver and/or a modem. I/O interface 626 may acquire encoded video data from the source device 610 or the storage medium/ server 640. Video decoder 624 may decode the encoded video data. Display device 622 may display the decoded video data to a user. Display device 622 may be integrated with the destination device 620, or may be external to destination device 620 which may be configured to interface with an external display device.

[00666] Video encoder 614 and video decoder 624 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding (VVC) standard, and other current and/or further standards.

[00667] FIG. 7 is a block diagram illustrating an example of video encoder 700, which may be video encoder 614 in the video coding system 600 illustrated in FIG. 6.

[00668] Video encoder 700 may be configured to perform any or all of the techniques of this disclosure. In the example of FIG. 7, video encoder 700 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of video encoder 700. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[00669] The functional components of video encoder 700 may include a partition unit 701, a prediction unit 702 which may include a mode selection unit 703, a motion estimation unit 704, a motion compensation unit 705 and an intra prediction unit 706, a residual generation unit 707, a transform unit 708, a quantization unit 709, an inverse quantization unit 710, an inverse transform unit 711, a reconstruction unit 712, a buffer 713, and an entropy encoding unit 714.

[00670] In other examples, video encoder 700 may include more, fewer, or different functional components. In an example, prediction unit 702 may include an intra block copy (IBC) unit. The IBC unit may perform prediction in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[00671] Furthermore, some components, such as motion estimation unit 704 and motion compensation unit 705 may be highly integrated, but are represented in the example of FIG. 7 separately for purposes of explanation.

[00672] Partition unit 701 may partition a picture into one or more video blocks. Video encoder 614 and video decoder 624 of FIG. 6 may support various video block sizes.

[00673] Mode selection unit 703 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra- or inter-coded block to a residual generation unit

707 to generate residual block data and to a reconstruction unit 712 to reconstruct the encoded block for use as a reference picture. In some examples, mode selection unit 703 may select a combination of intra and inter prediction (CIIP) mode in which the prediction is based on an inter prediction signal and an intra prediction signal. Mode selection unit 703 may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter-prediction.

**[00674]** To perform inter prediction on a current video block, motion estimation unit 704 may generate motion information for the current video block by comparing one or more reference frames from buffer 713 to the current video block. Motion compensation unit 705 may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from buffer 713 other than the picture associated with the current video block.

**[00675]** Motion estimation unit 704 and motion compensation unit 705 may perform different operations for a current video block, for example, depending on whether the current video block is in an I slice, a P slice, or a B slice. I-slices (or I-frames) are the least compressible but don't require other video frames to decode. S-slices (or P-frames) can use data from previous frames to decompress and are more compressible than I-frames. B-slices (or B-frames) can use both previous and forward frames for data reference to get the highest amount of data compression.

**[00676]** In some examples, motion estimation unit 704 may perform uni-directional prediction for the current video block, and motion estimation unit 704 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. Motion estimation unit 704 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. Motion estimation unit 704 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. Motion compensation unit 705 may generate the predicted video block of the current block based on the reference video block indicated by the motion information of the current video block.

**[00677]** In other examples, motion estimation unit 704 may perform bi-directional prediction for the current video block, motion estimation unit 704 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. Motion estimation unit 704 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. Motion estimation unit 704 may

output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. Motion compensation unit 705 may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.

**[00678]** In some examples, motion estimation unit 704 may output a full set of motion information for decoding processing of a decoder.

**[00679]** In some examples, motion estimation unit 704 may not output a full set of motion information for the current video. Rather, motion estimation unit 704 may signal the motion information of the current video block with reference to the motion information of another video block. For example, motion estimation unit 704 may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

**[00680]** In one example, motion estimation unit 704 may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder 624 that the current video block has the same motion information as another video block.

**[00681]** In another example, motion estimation unit 704 may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder 624 may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

**[00682]** As discussed above, video encoder 614 may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder 614 include advanced motion vector prediction (AMVP) and merge mode signaling.

**[00683]** Intra prediction unit 706 may perform intra prediction on the current video block. When intra prediction unit 706 performs intra prediction on the current video block, intra prediction unit 706 may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

**[00684]** Residual generation unit 707 may generate residual data for the current video block by subtracting (e.g., indicated by the minus sign) the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

[00685] In other examples, there may be no residual data for the current video block, for example in a skip mode, and residual generation unit 707 may not perform the subtracting operation.

[00686] Transform unit 708 may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

[00687] After transform unit 708 generates a transform coefficient video block associated with the current video block, quantization unit 709 may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[00688] Inverse quantization unit 710 and inverse transform unit 711 may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. Reconstruction unit 712 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the prediction unit 702 to produce a reconstructed video block associated with the current block for storage in the buffer 713.

[00689] After reconstruction unit 712 reconstructs the video block, loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[00690] Entropy encoding unit 714 may receive data from other functional components of the video encoder 700. When entropy encoding unit 714 receives the data, entropy encoding unit 714 may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[00691] FIG. 8 is a block diagram illustrating an example of video decoder 800, which may be video decoder 624 in the video coding system 600 illustrated in FIG. 6.

[00692] The video decoder 800 may be configured to perform any or all of the techniques of this disclosure. In the example of FIG. 8, the video decoder 800 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 800. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[00693] In the example of FIG. 8, video decoder 800 includes an entropy decoding unit 801, a motion compensation unit 802, an intra prediction unit 803, an inverse quantization unit 804, an inverse transformation unit 805, a reconstruction unit 806, and a buffer 807. Video decoder 800 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 614 (FIG. 6).

**[00694]** Entropy decoding unit 801 may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). Entropy decoding unit 801 may decode the entropy coded video data, and from the entropy decoded video data, motion compensation unit 802 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. Motion compensation unit 802 may, for example, determine such information by performing the AMVP and merge mode signaling.

**[00695]** Motion compensation unit 802 may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

**[00696]** Motion compensation unit 802 may use interpolation filters as used by video encoder 614 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. Motion compensation unit 802 may determine the interpolation filters used by video encoder 614 according to received syntax information and use the interpolation filters to produce predictive blocks.

**[00697]** Motion compensation unit 802 may use some of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter-encoded block, and other information to decode the encoded video sequence.

**[00698]** Intra prediction unit 803 may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. Inverse quantization unit 804 inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by entropy decoding unit 801. Inverse transform unit 805 applies an inverse transform.

**[00699]** Reconstruction unit 806 may sum the residual blocks with the corresponding prediction blocks generated by motion compensation unit 802 or intra prediction unit 803 to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in buffer 807, which provides reference blocks for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

**[00700]** FIG. 9 is a method 900 for coding video data according to an embodiment of the disclosure. The method 900 may be performed by a coding apparatus (e.g., an encoder) having

a processor and a memory. The method 900 may be implemented when using SEI messages to convey information in a bitstream.

**[00701]** In block 902, the coding apparatus uses a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element. The SDI SEI message is a type of SEI message like, for example, the SEI message in the bitstream 300 of FIG. 3. The SDI view identifier length minus L syntax element is a type of syntax element like, for example, the syntax elements 324 in the bitstream 300 of FIG. 3. The SEI message, including the SDI SEI message, may carry any of the elements of syntax disclosed herein.

**[00702]** In block 904, the coding apparatus converts between a video media file and the bitstream based on the SDI SEI message.

**[00703]** When implemented in an encoder, converting includes receiving a media file (e.g., a video unit) and encoding an SEI message into a bitstream. When implemented in a decoder, converting includes receiving the bitstream including the SEI message, and decoding the SEI message in the bitstream to generate the video media file.

**[00704]** In an embodiment, the SDI view identifier length minus L syntax element is configured to prevent a length of an SDI view identifier value syntax element, which specifies a view identifier of an i-th layer in a bitstream, from being zero. In an embodiment, L is equal to 1. In an embodiment, the SDI view identifier length minus L syntax element is designated `sdi_view_id_len_minus1`. In an embodiment, the SDI view identifier value syntax element is designated `sdi_view_id_val[ i ]`. In an embodiment, the SDI view identifier length minus L syntax element, plus one, specifies the length of the SDI view identifier value syntax element.

**[00705]** In an embodiment, the SDI view identifier length minus L syntax element is coded as an unsigned integer using N bits. By way of example, an unsigned integer is an integer (e.g., a whole number) that does not have a sign (e.g., positive or negative) associated therewith. In an embodiment, N is equal to 4.

**[00706]** In an embodiment, the SDI view identifier length minus L syntax element is coded as a fixed-pattern bitstring using N bits, a signed integer using N bits, a truncated binary, a signed integer K-th order Exp-Golomb-coded syntax element where K is equal to 0, or an unsigned integer M-th order Exp-Golomb-coded syntax element where M is equal to 0. A bitstring is an array data structure that compactly stores bits. A fixed-pattern bitstring is an array data structure having a fixed pattern. A signed integer is an integer (e.g., a whole number) that has a sign (e.g., positive or negative) associated therewith. Truncated binary, or truncated binary encoding, is an entropy encoding typically used for uniform probability distributions

with a finite alphabet. An exponential-Golomb code (Exp-Golomb code) is a type of universal code.

**[00707]** In an embodiment, the bitstream is a bitstream in scope. In an embodiment, the bitstream in scope is a sequence of access units (AUs) that consists, in decoding order, of an initial AU containing the SDI SEI message followed by zero or more subsequent AUs up to, but not including, any subsequent AU that contains another SDI SEI message.

**[00708]** In an embodiment, a multiview information SEI message and an auxiliary information SEI message are not present in a coded video sequence (CVS) unless the SDI SEI message is present in the CVS.

**[00709]** In an embodiment, the multiview information SEI message comprises a multiview acquisition information SEI message. In an embodiment, the auxiliary information SEI message comprises a depth representation information SEI message. In an embodiment, the auxiliary information SEI message comprises an alpha channel information SEI message.

**[00710]** In an embodiment, one or more of an SDI multiview information flag (e.g., `sdi_multiview_info` flag) and an SDI auxiliary information flag (e.g., `sdi_auxiliary_info` flag) are equal to 1 when the multiview information SEI message or the auxiliary information SEI message are present in the bitstream. A flag is a variable or single-bit syntax element that can take one of the two possible values: 0 and 1.

**[00711]** In an embodiment, the multiview information SEI message comprises a multiview acquisition information SEI message, and the multiview acquisition information SEI message is not scalable-nested. A scalable-nested SEI message is an SEI message within a scalable nesting SEI message. A scalable nesting SEI message is a message that contains a plurality of scalable-nested SEI messages that correspond to one or more output layer sets or one or more layers in a multi-layer bitstream.

**[00712]** In an embodiment, an SEI message in the bitstream and having a payload type equal to 179 is constrained from being included in a scalable nesting SEI message. In an embodiment, an SEI message in the bitstream and having a payload type equal to 3, 133, 179, 180, or 205 is constrained from being included in a scalable nesting SEI message.

**[00713]** In an embodiment, the method 900 may utilize or incorporate one or more of the features or processes of the other methods disclosed herein.

**[00714]** A listing of solutions preferred by some embodiments is provided next.

**[00715]** The following solutions show example embodiments of techniques discussed in the present disclosure (e.g., Example 1).

- [00716] 1. A method of video processing, comprising: performing a conversion between a video and a bitstream of the video; wherein the bitstream conforms to a format rule; wherein the format rule specifies that a syntax element indicates a length of view identifier syntax elements minus L, where L is an integer.
- [00717] 2. The method of claim 1, wherein the syntax element is coded as an unsigned integer using N bits.
- [00718] 3. The method of any of claims 1-2, wherein L is a positive integer.
- [00719] 4. The method of claim 1, wherein L=0, and wherein the syntax element is disallowed to have a zero value.
- [00720] 5. A method of video processing, comprising: performing a conversion between a video comprising multiple layers and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies that the bitstream includes an auxiliary layer that is associated with one or more associated layers of the video.
- [00721] 6. The method of claim 5, wherein the format rule further specifies whether or how the bitstream includes one or more syntax elements indicative of a relationship between the auxiliary layer and the one or more associated layers, wherein the one or more syntax elements are included in a scalability dimension supplemental enhancement information syntax structure.
- [00722] 7. The method of claim 6, wherein the format rule specifies that the one or more associated layers are indicated by corresponding layer identifiers (IDs).
- [00723] 8. The method of claim 6, wherein the format rule specifies that the one or more associated layers are indicated by corresponding layer indices.
- [00724] 9. The method of any of claims 5-8, wherein the format rule specifies that the bitstream includes one or more syntax elements indicating whether the auxiliary layer is applicable to the one or more associated layers.
- [00725] 10. The method of claim 9, wherein the one or more syntax elements comprise a syntax element indicating that the auxiliary layer is applicable to all of the one or more associated layers.
- [00726] 11. The method of claim 9, wherein the format rule specifies that a syntax element is included for each associated layer indicating whether the auxiliary layer is applicable to a corresponding associated layer.
- [00727] 12. The method of claim 11, wherein the syntax element indicates all primary layers associated with the auxiliary layer.

- [00728] 13. The method of claim 11, wherein the syntax element indicates all primary layers associated with the auxiliary layer and having a layer index smaller than that of the auxiliary layer.
- [00729] 14. The method of claim 11, wherein the syntax element indicates all primary layers associated with the auxiliary layer and having a layer index greater than that of the auxiliary layer.
- [00730] 15. The method of any of claims 11-14, wherein the syntax element is a flag.
- [00731] 16. The method of claim 6, wherein the format rule specifies that the bitstream does not include an explicit syntax element indicating applicability of the auxiliary layer to the one or more associated layers and the applicability is derived during the conversion.
- [00732] 17. The method of claim 16, wherein the format rule specifies that the associated layers for the auxiliary layers have a layer ID that is equal to a layer ID of the auxiliary layer plus  $N_1, N_2, \dots, N_k$ , where  $k$  is an integer and no two  $N_i$  are equal to each other for  $i = 1, \dots, k$ .
- [00733] 18. The method of claim 17, wherein  $k = 1$  and  $N_1$  is one of 1, -1, 2 or -2.
- [00734] 19. The method of claim 17, wherein  $k$  is greater than 1.
- [00735] 20. The method of claim 19, wherein  $k$  is equal to 2 and  $N_1 = 1, N_2 = 2$ .
- [00736] 21. The method of claim 5, wherein the format rule further specifies that the bitstream omits one or more syntax elements indicative of a relationship between the auxiliary layer and the one or more associated layers, and wherein the relationship is derived based on pre-determined rules.
- [00737] 22. The method of claim 5, wherein the format rule further specifies that the bitstream includes one or more syntax elements indicative of a relationship between the auxiliary layer and the one or more associated layers, wherein the one or more syntax elements are included in an auxiliary information supplemental enhancement information syntax structure.
- [00738] 23. The method of any of claims 5-22, wherein the format rule specifies that a syntax element is included in the bitstream indicative of a number of associated layers of auxiliary pictures of a layer.
- [00739] 24. The method of any of claims 5-22, wherein the format rule specifies that a syntax element is included in the bitstream indicative of a number of associated layers of auxiliary pictures of a layer or associated layers of auxiliary pictures in case that a condition is met.
- [00740] 25. The method of claim 24, wherein the condition comprises that an  $i$ -th layer in the bitstreamInScope includes auxiliary pictures.
- [00741] 26. A method of video processing, comprising: performing a conversion between a video comprising multiple video layers and a bitstream of the video, wherein the bitstream

conforms to a format rule, wherein the format rule specifies that a coded video sequence of the bitstream included a multiview supplemental enhancement information (SEI) message or an auxiliary information SEI message responsive to whether a scalability dimension information SEI message is included in a coded video sequence.

[00742] 27. The method of claim 26, wherein the format rule specifies that the multiview information SEI message refers to a multiview acquisition information SEI message.

[00743] 28. The method of any of claims 26-27, wherein the format rule specifies that the auxiliary information SEI message refers to a depth representation information SEI message or an alpha channel information SEI message.

[00744] 29. A method of video processing, comprising: performing a conversion between a video comprising multiple video layers and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies that responsive to a multiview or an auxiliary information supplemental enhancement information (SEI) message being present in the bitstream, at least one of a first flag indicating a presence of multiview information or a second flag indicating presence of auxiliary information in a scalability dimension information SEI message is equal to 1.

[00745] 30. A method of video processing, comprising: performing a conversion between a video comprising multiple video layers and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies that a multiview acquisition information supplemental enhancement information message included in the bitstream is not scalable-nested or included in a scalable nesting supplemental enhancement information message.

[00746] 31. The method of any of claims 1-30, wherein the conversion comprises generating the video from the bitstream or generating the bitstream from the video.

[00747] 32. A method of storing a bitstream on a computer-readable medium, comprising generating a bitstream according to a method recited in any one or more of claims 1-31 and storing the bitstream on the computer-readable medium.

[00748] 33. A computer-readable medium having a bitstream of a video stored thereon, the bitstream, when processed by a processor of a video decoder, causing the video decoder to generate the video, wherein the bitstream is generated according to a method recited in one or more of claims 1-31.

[00749] 34. A video decoding apparatus comprising a processor configured to implement a method recited in one or more of claims 1 to 31.

[00750] 35. A video encoding apparatus comprising a processor configured to implement a method recited in one or more of claims 1 to 31.

[00751] 36. A computer program product having computer code stored thereon, the code, when executed by a processor, causes the processor to implement a method recited in any of claims 1 to 31.

[00752] 37. A computer readable medium on which a bitstream complying to a bitstream format that is generated according to any of claims 1 to 31.

[00753] 38. A method, an apparatus, a bitstream generated according to a disclosed method or a system described in the present document.

[00754] The following documents may include additional details related to the techniques disclosed herein:

[00755] [1] ITU-T and ISO/IEC, “High efficiency video coding”, Rec. ITU-T H.265 | ISO/IEC 23008-2 (in force edition).

[00756] [2] J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, “Algorithm description of Joint Exploration Test Model 7 (JEM7),” JVET-G1001, Aug. 2017.

[00757] [3] Rec. ITU-T H.266 | ISO/IEC 23090-3, “Versatile Video Coding”, 2020.

[00758] [4] B. Bross, J. Chen, S. Liu, Y.-K. Wang (editors), “Versatile Video Coding (Draft 10),” JVET-S2001.

[00759] [5] Rec. ITU-T Rec. H.274 | ISO/IEC 23002-7, “Versatile Supplemental Enhancement Information Messages for Coded Video Bitstreams”, 2020.

[00760] [6] J. Boyce, V. Drugeon, G. Sullivan, Y.-K. Wang (editors), “Versatile supplemental enhancement information messages for coded video bitstreams (Draft 5),” JVET-S2007.

[00761] The disclosed and other solutions, examples, embodiments, modules and the functional operations described in this document can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this document and their structural equivalents, or in combinations of one or more of them. The disclosed and other embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them. The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of

example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

**[00762]** A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

**[00763]** The processes and logic flows described in this document can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

**[00764]** Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example

semiconductor memory devices, e.g., erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and compact ?disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

**[00765]** While this patent document contains many specifics, these should not be construed as limitations on the scope of any subject matter or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular techniques. Certain features that are described in this patent document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

**[00766]** Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described in this patent document should not be understood as requiring such separation in all embodiments.

**[00767]** Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this patent document.

## CLAIMS

What is claimed is:

1. A method of processing video data, comprising:  
using a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and  
performing a conversion between a video media file and the bitstream based on the SDI SEI message.
2. The method of claim 1, wherein the SDI view identifier length minus L syntax element is configured to prevent a length of an SDI view identifier value syntax element, which specifies a view identifier of an i-th layer in a bitstream, from being zero.
3. The method of any of claims 1-2, wherein L is equal to 1.
4. The method of any of claims 1-3, wherein the SDI view identifier length minus L syntax element is designated `sdi_view_id_len_minus1`.
5. The method of any of claims 1-4, wherein the SDI view identifier value syntax element is designated `sdi_view_id_val[ i ]`.
6. The method of any of claims 1-5, wherein the SDI view identifier length minus L syntax element, plus one, specifies the length of the SDI view identifier value syntax element.
7. The method of any of claims 1-6, wherein the SDI view identifier length minus L syntax element is coded as an unsigned integer using N bits.
8. The method of any of claims 1-7, wherein N is equal to 4.
9. The method of any of claims 1-8, wherein the SDI view identifier length minus L syntax element is coded as a fixed-pattern bitstring using N bits, a signed integer using N bits, a truncated binary, a signed integer K-th order Exp-Golomb-coded syntax element where K is equal to 0, or an unsigned integer M-th order Exp-Golomb-coded syntax element where M is equal to 0.

10. The method of any of claims 1-9, wherein the bitstream is a bitstream in scope.
11. The method of any of claims 1-10, wherein a multiview information SEI message and an auxiliary information SEI message are not present in a coded video sequence (CVS) unless the SDI SEI message is present in the CVS.
12. The method of any of claims 10-11, wherein the multiview information SEI message comprises a multiview acquisition information SEI message.
13. The method of any of claims 10-12, wherein the auxiliary information SEI message comprises a depth representation information SEI message.
14. The method of any of claims 10-13, wherein the auxiliary information SEI message comprises an alpha channel information SEI message.
15. The method of any of claims 1-14, wherein one or more of an SDI multiview information flag and an SDI auxiliary information flag are equal to 1 when the multiview information SEI message or the auxiliary information SEI message are present in the bitstream.
16. The method of any of claims 1-15, wherein the multiview information SEI message comprises a multiview acquisition information SEI message, and wherein the multiview acquisition information SEI message is not scalable-nested.
17. The method of any of claims 1-16, wherein an SEI message in the bitstream and having a payload type equal to 179 is constrained from being included in a scalable nesting SEI message.
18. The method of any of claims 1-17, wherein an SEI message in the bitstream and having a payload type equal to 3, 133, 179, 180, or 205 is constrained from being included in a scalable nesting SEI message.
19. The method of claim 1, wherein the conversion comprises encoding the video media file into the bitstream.

20. The method of claim 1, wherein the conversion includes decoding the bitstream to obtain the video media file.
21. An apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor cause the processor to:
- use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and
  - convert between a video media file and the bitstream based on the SDI SEI message.
22. A non-transitory computer readable medium comprising a computer program product for use by a coding apparatus, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium that, when executed by one or more processors, cause the coding apparatus to:
- use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and
  - convert between a video media file and the bitstream based on the SDI SEI message.
23. A non-transitory computer-readable storage medium storing instructions that cause a processor to:
- use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and
  - convert between a video media file and the bitstream based on the SDI SEI message.
24. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises:
- use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element; and
  - convert between a video media file and the bitstream based on the SDI SEI message.
25. A method for storing bitstream of a video, comprising:
- use a scalability dimension information (SDI) supplemental enhancement information (SEI) message to indicate an SDI view identifier length minus L syntax element;

generating the bitstream based on the SDI SEI message; and  
storing the bitstream in a non-transitory computer-readable recording medium.

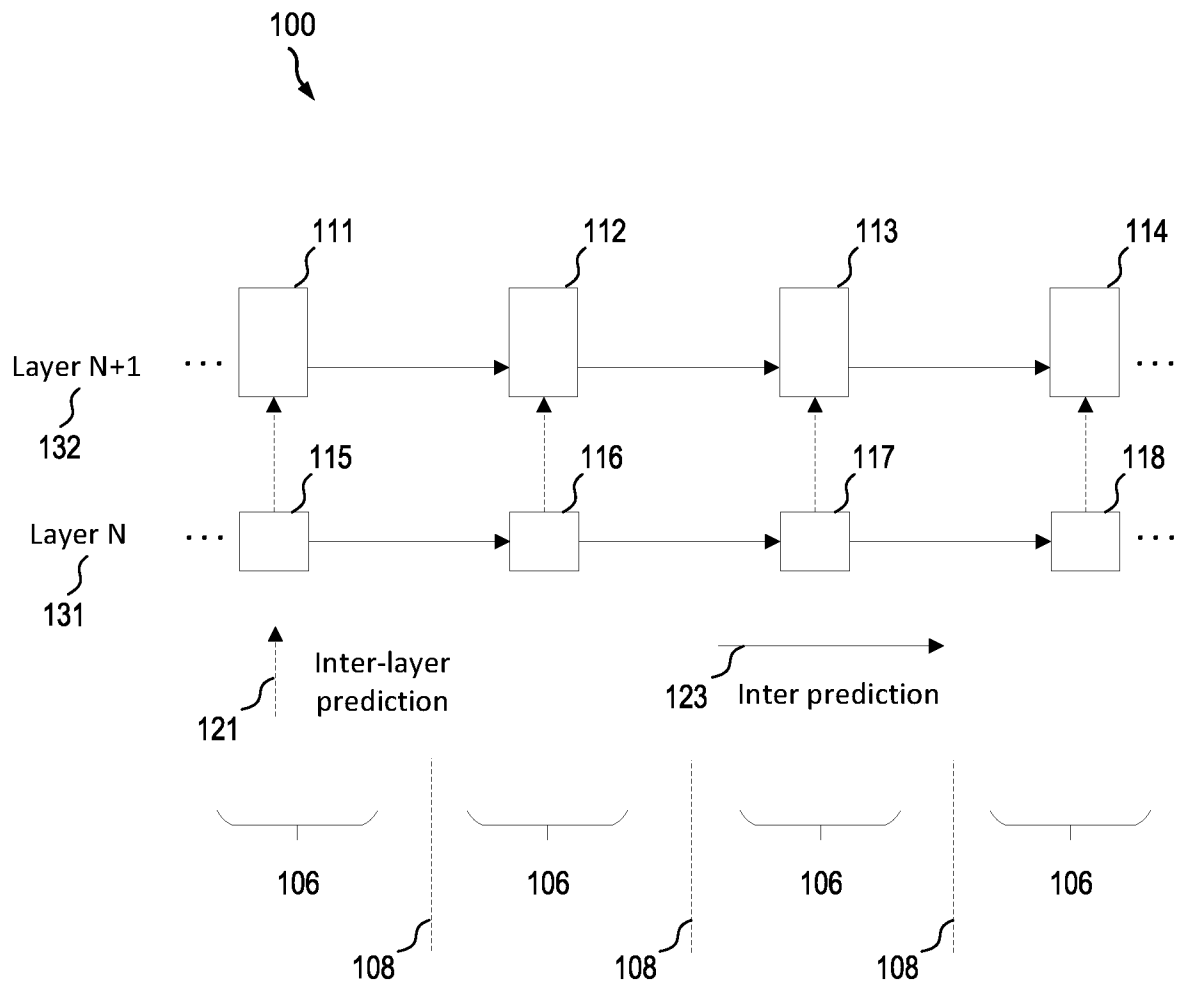


FIG. 1

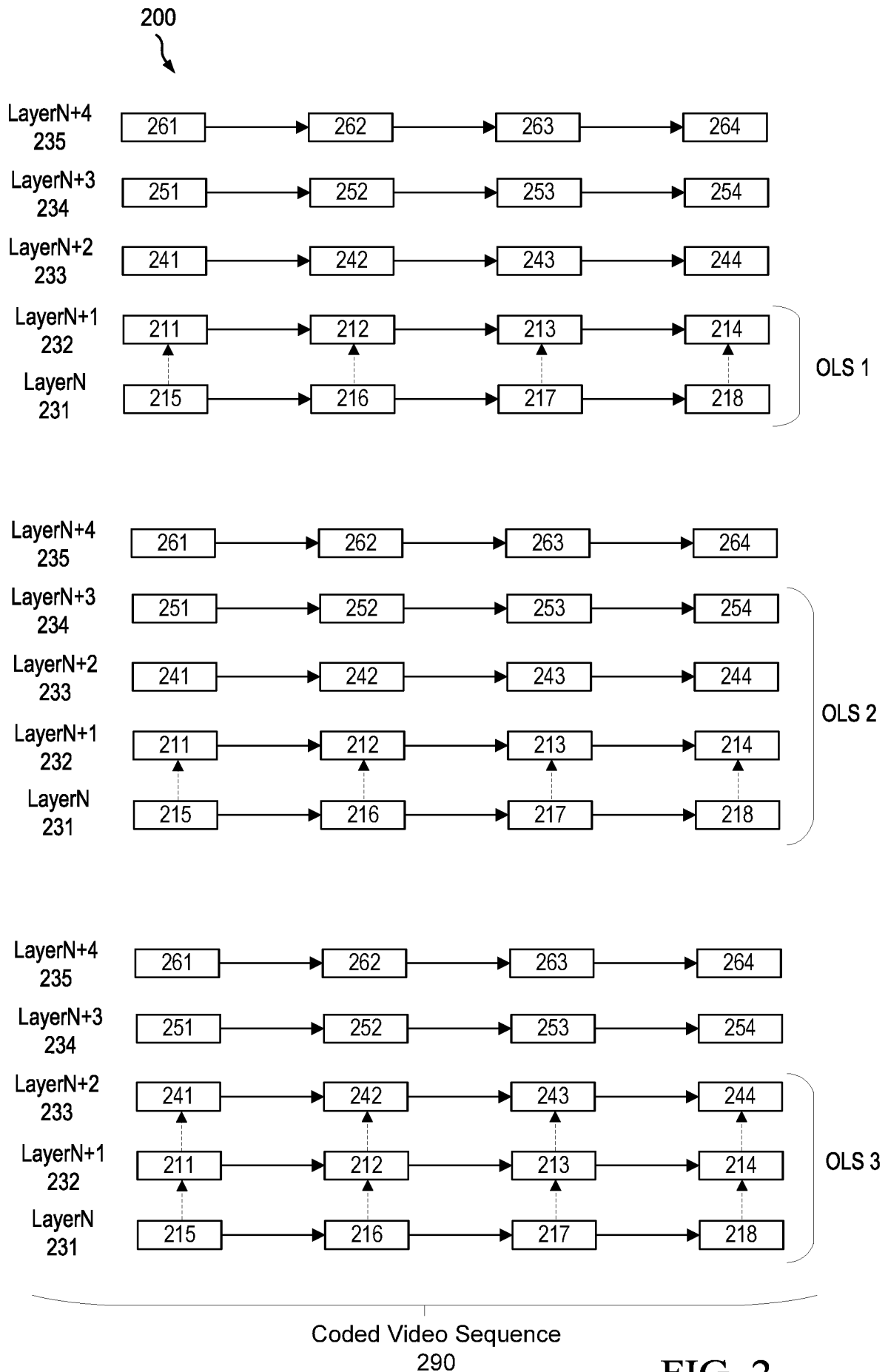


FIG. 2

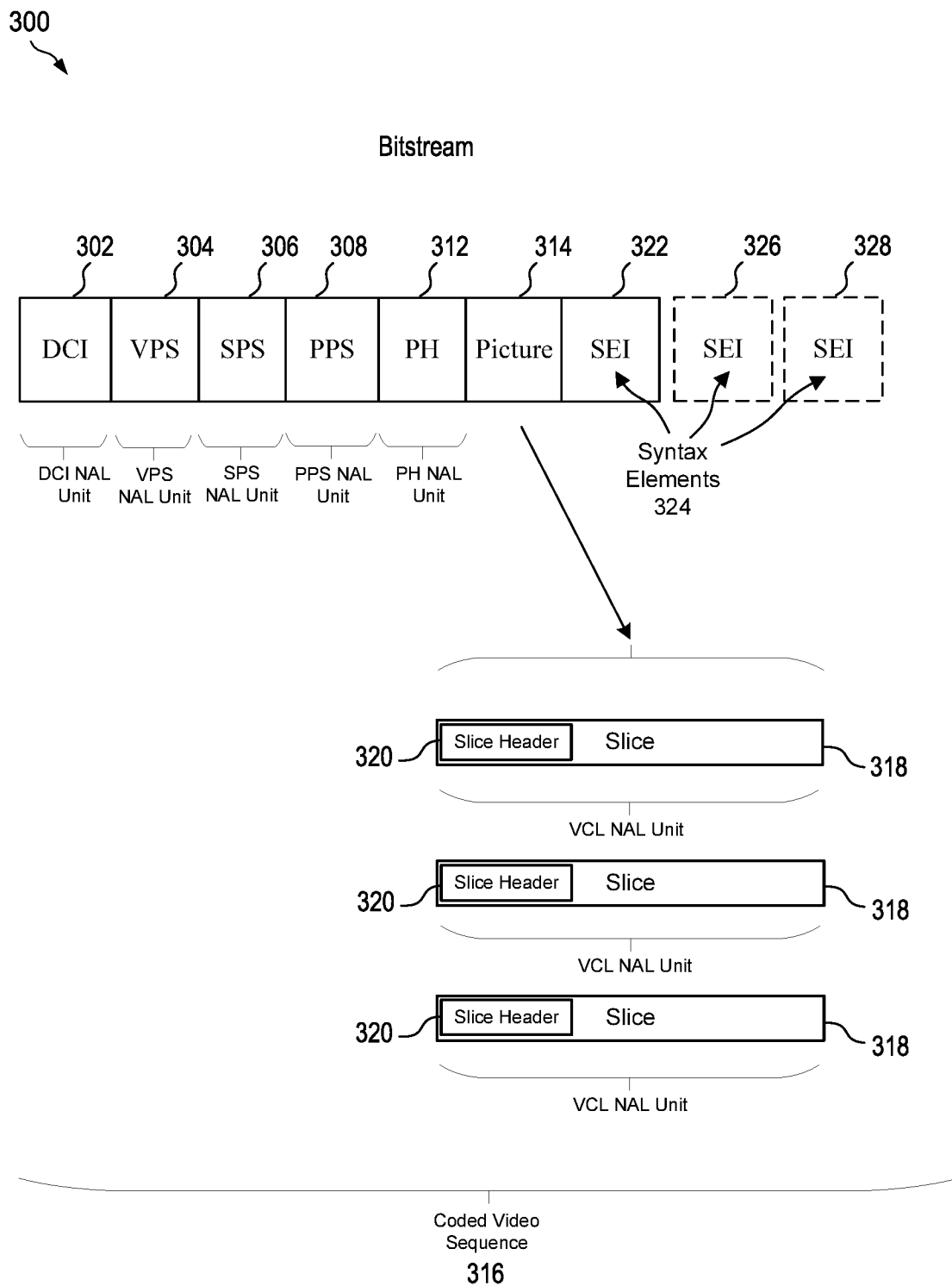


FIG. 3

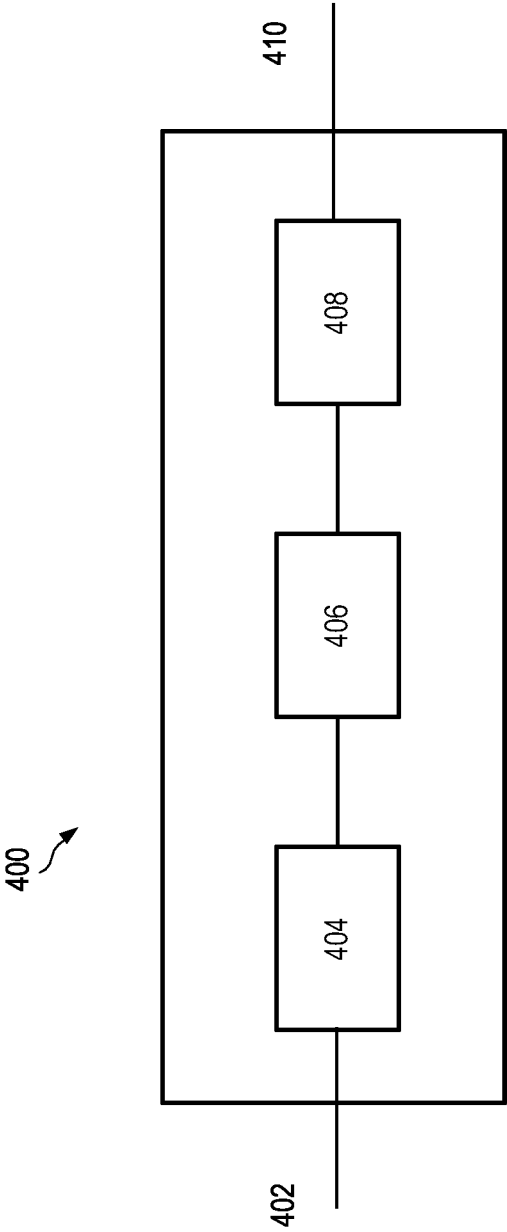


FIG. 4

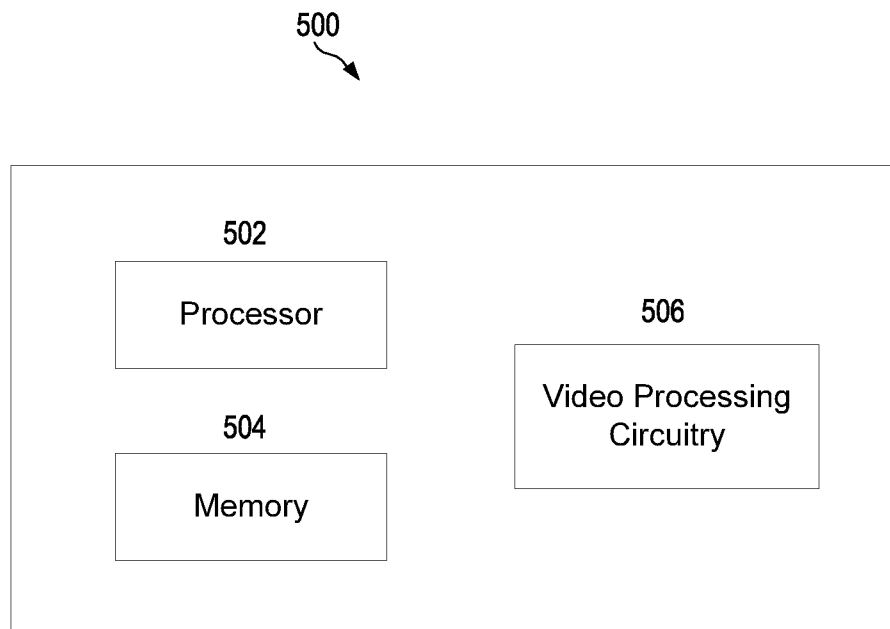


FIG. 5

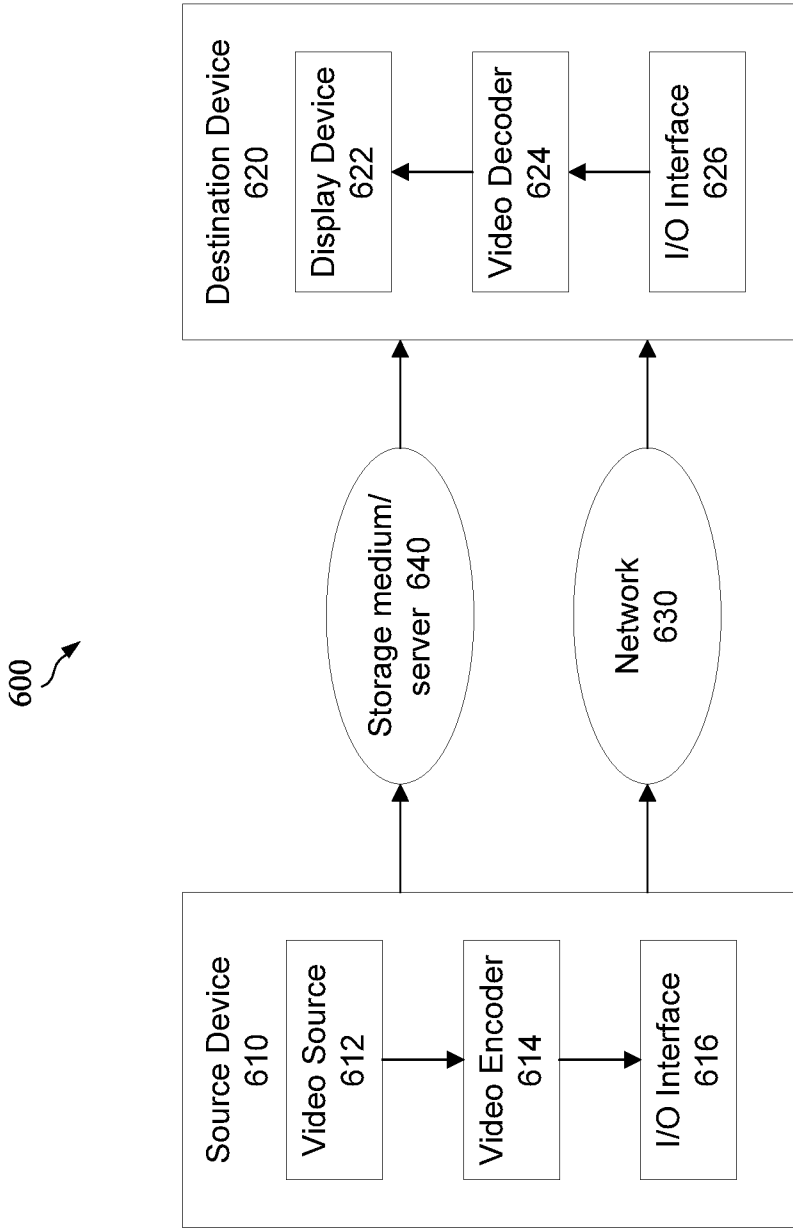


FIG. 6

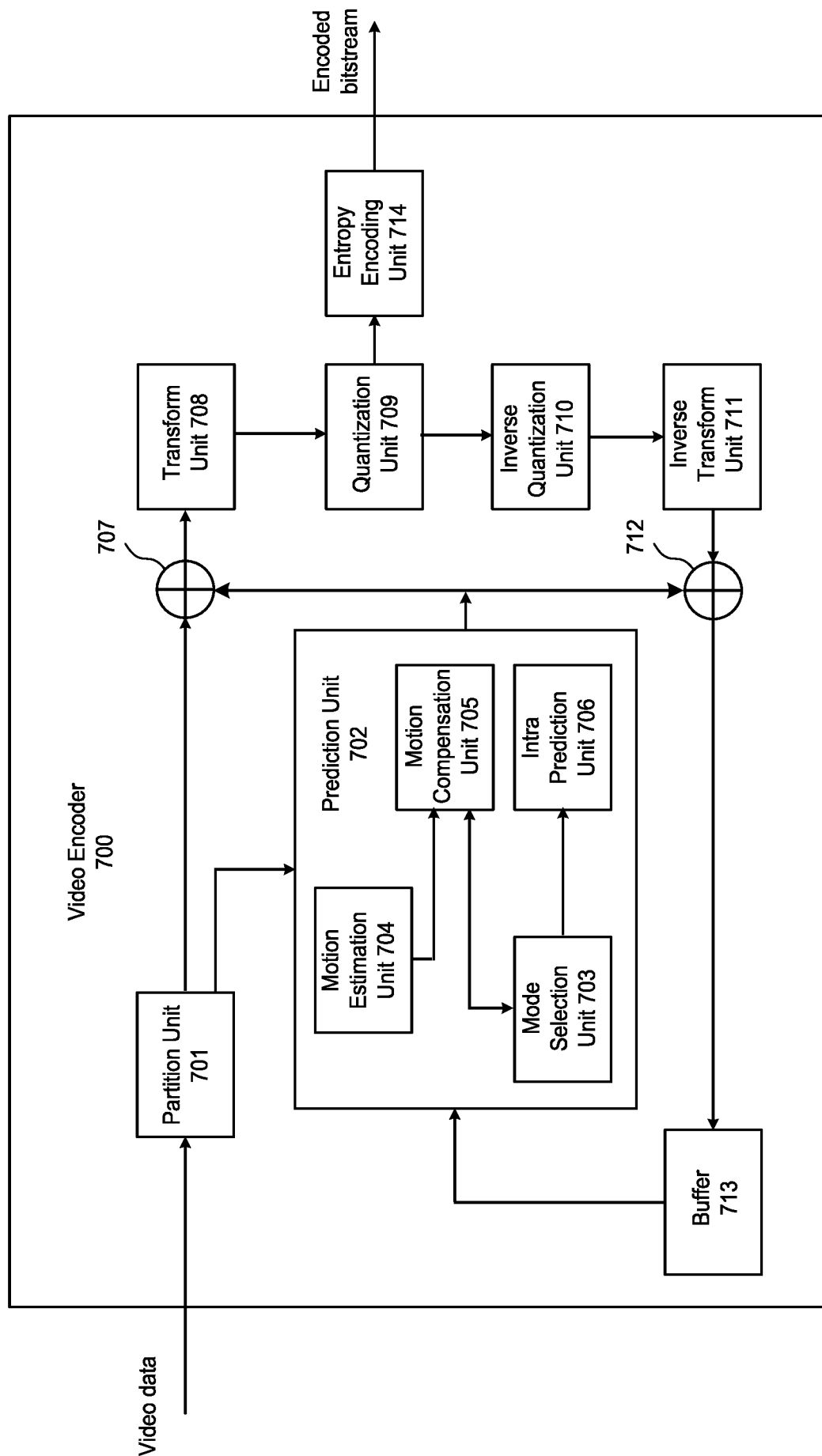


FIG. 7

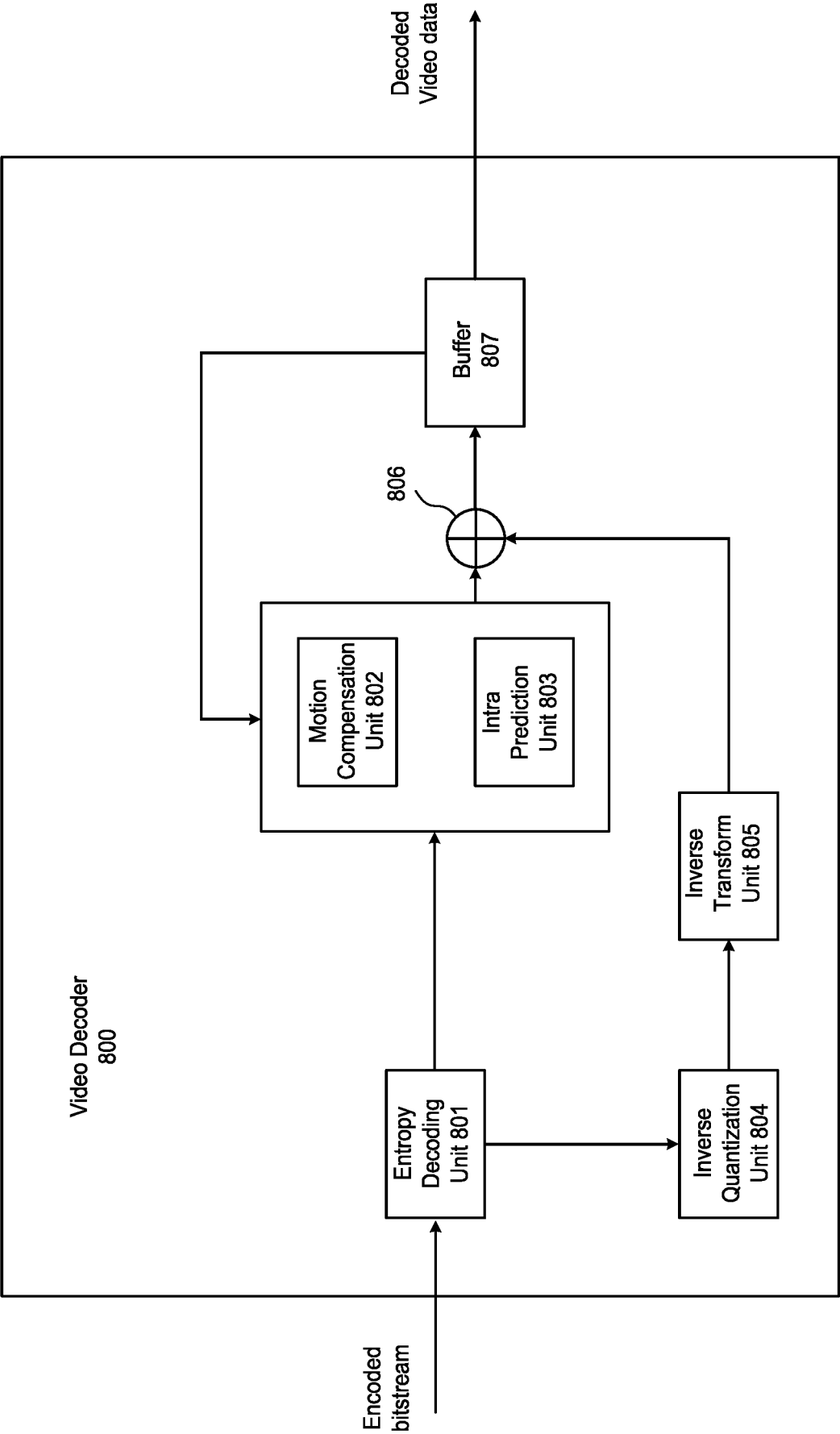


FIG. 8

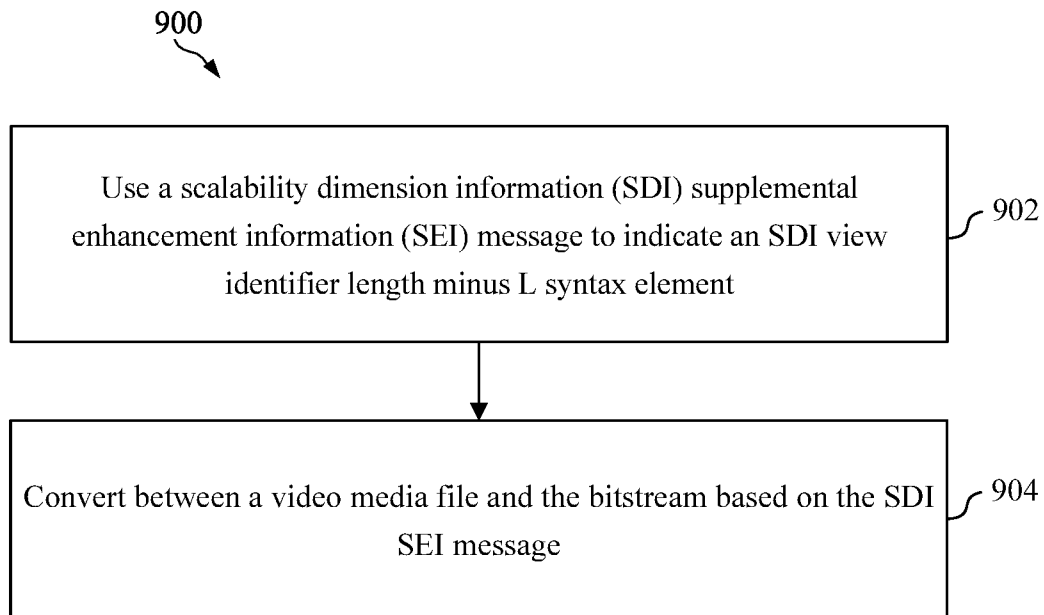


FIG. 9

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/084992

**A. CLASSIFICATION OF SUBJECT MATTER**

H04N 19/30(2014.01)i; H04N 19/70(2014.01)i; H04N 19/00(2014.01)i

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)

CNTXT; ENTXTC; CJFD; JNET: video, scalability, dimension, SDI, supplemental, enhancement, SEI, identifier, ID, layer, bitstream

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                   | Relevant to claim No. |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| A         | "Ye-Kui Wang et al."<br><i>AHG9/AHG8: Scalability dimension SEI message</i> ,<br>No. 01 Oct.2020(01.10.2020), 30 September 2020 (2020-09-30),<br>abstract, introduction and proposal | 1-25                  |
| A         | WO 2020228691 A1 (BEIJING BYTEDANCE NETWORK TECH CO., LTD. et al.) 19<br>November 2020 (2020-11-19)<br>the whole document                                                            | 1-25                  |
| A         | JP 2015177318 A (SHARP KK.) 05 October 2015 (2015-10-05)<br>the whole document                                                                                                       | 1-25                  |
| A         | CN 105531999 A (NOKIA CORP.) 27 April 2016 (2016-04-27)<br>the whole document                                                                                                        | 1-25                  |
| A         | CN 105794212 A (QUALCOMM INC.) 20 July 2016 (2016-07-20)<br>the whole document                                                                                                       | 1-25                  |
| A         | CN 104798371 A (SHARP KK.) 22 July 2015 (2015-07-22)<br>the whole document                                                                                                           | 1-25                  |



Further documents are listed in the continuation of Box C.



See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"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)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"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

"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

"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

"&" document member of the same patent family

Date of the actual completion of the international search

08 June 2022

Date of mailing of the international search report

15 June 2022

Name and mailing address of the ISA/CN

National Intellectual Property Administration, PRC  
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing  
100088, China

Facsimile No. (86-10)62019451

Authorized officer

MA,Hui

Telephone No. 86-(010)-62411524

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/084992

### C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                           | Relevant to claim No. |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| PX        | "Yang Wang et al."<br><i>AHG9: On the scalability dimension information SEI message</i> ,<br>No. 13 Apr.2021(13.04.2021), 13 April 2021 (2021-04-13),<br>abstract, introduction and proposal | 1-25                  |
| <hr/>     |                                                                                                                                                                                              |                       |

**INTERNATIONAL SEARCH REPORT**  
**Information on patent family members**

International application No.

**PCT/CN2022/084992**

| Patent document<br>cited in search report |            |    | Publication date<br>(day/month/year) | Patent family member(s) |             |    | Publication date<br>(day/month/year) |
|-------------------------------------------|------------|----|--------------------------------------|-------------------------|-------------|----|--------------------------------------|
| WO                                        | 2020228691 | A1 | 19 November 2020                     | CN                      | 113875246   | A  | 31 December 2021                     |
|                                           |            |    |                                      | US                      | 2022060712  | A1 | 24 February 2022                     |
|                                           |            |    |                                      | EP                      | 3954124     | A1 | 16 February 2022                     |
|                                           |            |    |                                      | CN                      | 113875250   | A  | 31 December 2021                     |
|                                           |            |    |                                      | WO                      | 2020228692  | A1 | 19 November 2020                     |
|                                           |            |    |                                      | KR                      | 20220006048 | A  | 14 January 2022                      |
| JP                                        | 2015177318 | A  | 05 October 2015                      | None                    |             |    |                                      |
| CN                                        | 105531999  | A  | 27 April 2016                        | KR                      | 20160030560 | A  | 18 March 2016                        |
|                                           |            |    |                                      | TW                      | 201511531   | A  | 16 March 2015                        |
|                                           |            |    |                                      | WO                      | 2015004606  | A1 | 15 January 2015                      |
|                                           |            |    |                                      | US                      | 2015016531  | A1 | 15 January 2015                      |
|                                           |            |    |                                      | EP                      | 3020186     | A1 | 18 May 2016                          |
|                                           |            |    |                                      | US                      | 2020228827  | A1 | 16 July 2020                         |
|                                           |            |    |                                      | US                      | 2022086480  | A1 | 17 March 2022                        |
|                                           |            |    |                                      |                         |             |    |                                      |
| CN                                        | 105794212  | A  | 20 July 2016                         | JP                      | 2016538760  | A  | 08 December 2016                     |
|                                           |            |    |                                      | MX                      | 2016005100  | A  | 03 August 2016                       |
|                                           |            |    |                                      | EP                      | 3061251     | A1 | 31 August 2016                       |
|                                           |            |    |                                      | WO                      | 2015061585  | A1 | 30 April 2015                        |
|                                           |            |    |                                      | KR                      | 20160075586 | A  | 29 June 2016                         |
|                                           |            |    |                                      | US                      | 2015110167  | A1 | 23 April 2015                        |
| CN                                        | 104798371  | A  | 22 July 2015                         | EP                      | 2923488     | A1 | 30 September 2015                    |
|                                           |            |    |                                      | ES                      | 2878143     | T3 | 18 November 2021                     |
|                                           |            |    |                                      | CN                      | 109257607   | A  | 22 January 2019                      |
|                                           |            |    |                                      | WO                      | 2014080630  | A1 | 30 May 2014                          |
|                                           |            |    |                                      | EP                      | 3923584     | A4 | 15 December 2021                     |
|                                           |            |    |                                      | PT                      | 2923488     | T  | 09 July 2021                         |
|                                           |            |    |                                      | PL                      | 2923488     | T3 | 02 November 2021                     |
|                                           |            |    |                                      | HK                      | 1212836     | A1 | 08 July 2016                         |