 ISO/IEC JTC 1/SC 29/WG 5 N0031

**ISO/IEC JTC 1/SC 29/WG 5**

**MPEG Joint Video Coding Team(s) with ITU-T SG 16**

**Convenorship: DE**

**Document type:** General

**Title:** Working Draft 2 of ISO/IEC 23002-7 Amd.1 Additional SEI messages

**Status:** Approved

**Date of document:** 2021-01-15

**Source:** ISO/IEC JTC 1/SC 29/WG 5

**Expected action:** Info

**Action due date:** None

**No. of pages:** 20 (without cover page)

**Email of Convenor:** ohm @ ient . rwth-aachen . de

**Committee URL:** https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg5

|  |  |
| --- | --- |
| **Joint Video Experts Team (JVET)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29**  21st Meeting, by teleconference, 6–15 Jan. 2021 | Document: JVET-U2006-v1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Additional SEI messages for VSEI (Draft 2)** | | |
| *Status:* | Output document approved by JVET | | |
| *Purpose:* | Draft text | | |
| *Author(s) or Contact(s):* | Jill Boyce  Ye-Kui Wang | Email: | jill.boyce@intel.com yekui.wang@bytedance.com |
| *Source:* | Editors | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This document contains the draft text for changes to the versatile supplemental enhancement information messages for coded video bitstreams (VSEI) standard (Rec. ITU-T H.274 | ISO/IEC 23002-7), to specify additional SEI messages, including the annotated regions SEI message, the alpha channel information SEI message, the depth representation information SEI message, the multiview acquisition information SEI message, the scalability dimension information SEI message, and the extended dependent random access point (DRAP) indication SEI message. The draft text also includes text changes for some technical corrections and editorial improvements.

Draft 2 incorporated items:

* Addition of the alpha channel information SEI message, the depth representation information SEI message, the multiview acquisition information SEI message, and the scalability dimension information SEI message (JVET-U0082)
* Addition of the extended DRAP indication SEI message (JVET-U0084)
* Some minor editorial corrections and improvements (JVET-U0086)
* Fix for ticket [#1412](https://jvet.hhi.fraunhofer.de/trac/vvc/ticket/1421): change the coding of ffi\_display\_elemental\_periods\_minus1 from u(4) to u(8)

**Changes to the specification text:**

*Replace 2.3 with the following:*

* 1. **Additional references**

– ISO 11664-1 (in force), *Colorimetry – Part 1: CIE standard colorimetric observers*.

– ISO/IEC 11578:1996, *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC)*.

– Recommendation ITU-T T.35:2000, *Procedure for the allocation of ITU-T defined codes for non standard facilities.*

– IETF RFC 1321 (in force), *The MD5 Message-Digest Algorithm.*

– IETF RFC 5646 (in force), *Tags for Identifying Languages.*

*In clause 6.1, make the following changes:*

**6.1 General**

...

Technical specifications that reference this Specification for carrying VUI parameters syntax structure shall specify a container to carry the data of the VUI parameters syntax structure and to identify the length in bits of the VUI parameters syntax structure, e.g., the vui\_payload( ) syntax structure specified in Rec. ITU-T H.266 | ISO/IEC 23090-3. The design of the container ~~needs to~~ should provide the ability to detect the number of bits in the vui\_parameters( ) syntax structure and to allow the number of bits to be increased in future versions of this Specification, thus enabling this Specification to provide extensibility by directly appending additional syntax elements to the end of the vui\_parameters( ) syntax structure in future versions of this Specification. The syntax of the container of the vui\_parameters( ) syntax structure is outside the scope of this Specification.

Technical specifications that reference this Specification for carrying SEI messages shall specify a way to carry the payload syntax of each specified SEI message, to identify which SEI message is conveyed, and to identify the length in bits of the SEI message syntax structure, e.g., the sei\_payload( ) syntax structure specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 and Rec. ITU-T H.265 | ISO/IEC 23008-2. The design of the container ~~needs to~~ should provide the ability to detect the number of bits in an SEI message and to allow the number of bits to be increased in future versions of this Specification, thus enabling this Specification to provide extensibility by directly appending additional syntax elements to the end of the SEI message syntax structure in future versions of this Specification. The syntax of the container of the SEI messages as well as the method of identifying which SEI message is outside the scope of this Specification.

...

*Replace 6.3 with the following:*

**6.3 Specification of syntax functions and descriptors**

The functions presented in this clause are used in the syntactical description. These functions are expressed in terms of the value of the VUI parameters syntax or an SEI message syntax data pointer that indicates the position of the next bit to be read by the decoding process from the syntax structure.

read\_bits( n ) reads the next n bits from the syntax structure and advances the data pointer by n bit positions. When n is equal to 0, read\_bits( n ) is specified to return a value equal to 0 and to not advance the data pointer.

The following descriptors specify the parsing process of each syntax element:

– b(8): byte having any pattern of bit string (8 bits). The parsing process for this descriptor is specified by the return value of the function read\_bits( 8 ).

– f(n): fixed-pattern bit string using n bits written (from left to right) with the left bit first. The parsing process for this descriptor is specified by the return value of the function read\_bits( n ).

– i(n): signed integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function read\_bits( n ) interpreted as a two's complement integer representation with most significant bit written first.

– se(v): signed integer 0-th order Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in clause 9 with the order k equal to 0.

–– st(v): null-terminated string encoded as universal coded character set (UCS) transmission format-8 (UTF-8) characters as specified in ISO/IEC 10646. The parsing process is specified as follows: st(v) begins at a byte-aligned position in the bitstream and reads and returns a series of bytes from the bitstream, beginning at the current position and continuing up to but not including the next byte-aligned byte that is equal to 0x00, and advances the bitstream pointer by ( stringLength + 1 ) \* 8 bit positions, where stringLength is equal to the number of bytes returned.

NOTE – The st(v) syntax descriptor is only used in this Specification when the current position in the bitstream is a byte-aligned position.

u(n): unsigned integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function read\_bits( n ) interpreted as a binary representation of an unsigned integer with most significant bit written first.

– ue(v): unsigned integer 0-th order Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in clause 9 with the order k equal to 0.

*In clauses 8.16.1. change the coding of the ffi\_display\_elemental\_periods\_minus1 syntax element from u(4) to u(8).*

*Add clauses 8.18 to 8.22 as follows:*

* 1. **Annotated regions SEI message**

8.18.1 Annotated regions SEI message syntax

|  |  |  |
| --- | --- | --- |
| annotated\_regions( payloadSize ) { | **C** | **Descriptor** |
| **ar\_cancel\_flag** | 5 | u(1) |
| if(!ar\_cancel\_flag) { |  |  |
| **ar\_not\_optimized\_for\_viewing\_flag** | 5 | u(1) |
| **ar\_true\_motion\_flag** | 5 | u(1) |
| **ar\_occluded\_object\_flag** | 5 | u(1) |
| **ar\_partial\_object\_flag\_present\_flag** | 5 | u(1) |
| **ar\_object\_label\_present\_flag** | 5 | u(1) |
| **ar\_object\_confidence\_info\_present\_flag** | 5 | u(1) |
| if( ar\_object\_confidence\_info\_present\_flag ) |  |  |
| **ar\_object\_confidence\_length\_minus1** | 5 | u(4) |
| if( ar\_object\_label\_present\_flag ) { |  |  |
| **ar\_object\_label\_language\_present\_flag** | 5 | u(1) |
| if( ar\_object\_label\_language\_present\_flag ) { |  |  |
| while( !byte\_aligned( ) ) |  |  |
| **ar\_bit\_equal\_to\_zero** /\* equal to 0 \*/ | 5 | f(1) |
| **ar\_object\_label\_language** | 5 | st(v) |
| } |  |  |
| **ar\_num\_label\_updates** | 5 | ue(v) |
| for( i = 0; i < ar\_num\_ label\_updates; i++ ) { |  |  |
| **ar\_label\_idx**[ i ] | 5 | ue(v) |
| **ar\_label\_cancel\_flag** | 5 | u(1) |
| LabelAssigned[ ar\_label\_idx[ i ] ] = !ar\_label\_cancel\_flag |  |  |
| if( !ar\_label\_cancel\_flag ) { |  |  |
| while( !byte\_aligned( ) ) |  |  |
| **ar\_bit\_equal\_to\_zero** /\* equal to 0 \*/ | 5 | f(1) |
| **ar\_label**[ ar\_label\_idx[ i ] ] | 5 | st(v) |
| } |  |  |
| } |  |  |
| } |  |  |
| **ar\_num\_object\_updates** | 5 | ue(v) |
| for( i = 0; i  <=  ar\_num\_object\_updates; i++ ) { |  |  |
| **ar\_object\_idx**[ i ] | 5 | ue(v) |
| **ar\_object\_cancel\_flag** | 5 | u(1) |
| ObjectTracked[ ar\_object\_idx[ i ] ] = !ar\_object\_cancel\_flag |  |  |
| if( !ar\_object\_cancel\_flag ) { |  |  |
| if( ar\_object\_label\_present\_flag ) { |  |  |
| **ar\_object\_label\_update\_flag** | 5 | u(1) |
| if( ar\_object\_label\_update\_flag ) |  |  |
| **ar\_object\_label\_idx**[ ar\_object\_idx[ i ] ] | 5 | ue(v) |
| } |  |  |
| **ar\_bounding\_box\_update\_flag** | 5 | u(1) |
| if( ar\_bounding\_box\_update\_flag ) { |  |  |
| **ar\_bounding\_box\_cancel\_flag** | 5 | u(1) |
| ObjectBoundingBoxAvail[ ar\_object\_idx[ i ] ] = !ar\_bounding\_box\_cancel\_flag |  |  |
| if( !ar\_bounding\_box\_cancel\_flag ) { |  |  |
| **ar\_bounding\_box\_top[** ar\_object\_idx[ i ] ] | 5 | u(16) |
| **ar\_bounding\_box\_left**[ ar\_object\_idx[ i ] ] | 5 | u(16) |
| **ar\_bounding\_box\_width**[ ar\_object\_idx[ i ] ] | 5 | u(16) |
| **ar\_bounding\_box\_height**[ ar\_object\_idx[ i ] ] | 5 | u(16) |
| if( ar\_partial\_object\_flag\_present\_flag ) |  |  |
| **ar\_partial\_object\_flag**[ ar\_object\_idx[ i ] ] | 5 | u(1) |
| if( ar\_object\_confidence\_info\_present\_flag ) |  |  |
| **ar\_object\_confidence**[ ar\_object\_idx[ i ] ] | 5 | u(v) |
| } |  |  |
| } |  |  |
| } |  |  |
| } |  |  |
| } |  |  |
| } |  |  |

8.18.2 Annotated regions SEI message semantics

The annotated regions SEI message carries parameters that identify annotated regions using bounding boxes representing the size and location of identified objects.

Use of this SEI message requires the definition of the following variables:

- A cropped picture width and picture height in units of luma samples, denoted herein by CroppedWidth and CroppedHeight, respectively.

- A chroma sub-sampling width and height, SubWidthC and SubHeightC, respectively.

- A conformance cropping window left offset, ConfWinLeftOffset

- A conformance cropping window top offset, ConfWinTopOffset

**ar\_cancel\_flag** equal to 1 indicates that the annotated regions SEI message cancels the persistence of any previous annotated regions SEI message that is associated with one or more layers to which the annotated regions SEI message applies. ar\_cancel\_flag equal to 0 indicates that annotated regions information follows.

When ar\_cancel\_flag equal to 1 or a new CVS of the current layer begins, the variables LabelAssigned[ i ], ObjectTracked[ i ], and ObjectBoundingBoxAvail are set equal to 0 for i in the range of 0 to 255, inclusive.

**ar\_not\_optimized\_for\_viewing\_flag** equal to 1 indicates that the decoded pictures that the annotated regions SEI message applies to are not optimized for user viewing, but rather are optimized for some other purpose such as algorithmic object classification performance. ar\_not\_optimized\_for\_viewing\_flagequal to 0 indicates that the decoded pictures that the annotated regions SEI message applies to may or may not be optimized for user viewing.

**ar\_true\_motion\_flag** equal to 1 indicates that the motion information in the coded pictures that the annotated regions SEI message applies to was selected with a goal of accurately representing object motion for objects in the annotated regions. ar\_true\_motion\_flag equal to 0 indicates that the motion information in the coded pictures that the annotated regions SEI message applies to may or may not be selected with a goal of accurately representing object motion for objects in the annotated regions.

**ar\_occluded\_object\_flag** equal to 1 indicates that the ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ], and ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] syntax elements each represent the size and location of an object or a portion of an object that may not be visible or may be only partially visible within the cropped decoded picture. ar\_occluded\_object\_flagequal to 0 indicates that the ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ], and ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] syntax elements represent the size and location of an object that is entirely visible within the cropped decoded picture. It is a requirement of bitstream conformance that the value of ar\_occluded\_object\_flag shall be the same for all annotated\_regions( ) syntax structures within a CVS.

**ar\_partial\_object\_flag\_present\_flag** equal to 1 indicates that ar\_partial\_object\_flag[ ar\_object\_idx[ i ] ] syntax elements are present. ar\_partial\_object\_flag\_present\_flag equal to 0 indicates that ar\_partial\_object\_flag[ ar\_object\_idx[ i ] ] syntax elements are not present. It is a requirement of bitstream conformance that the value of ar\_partial\_object\_flag\_present\_flag shall be the same for all annotated\_regions( ) syntax structures within a CVS.

**ar\_object\_label\_present\_flag** equal to 1 indicates that label information corresponding to objects in the annotated regions is present. ar\_object\_label\_present\_flag equal to 0 indicates that label information corresponding to the objects in the annotated regions is not present.

**ar\_object\_confidence\_info\_present\_flag** equal to 1 indicates that ar\_object\_confidence[ ar\_object\_idx[ i ] ] syntax elements are present. ar\_object\_confidence\_info\_present\_flag equal to 0 indicates that ar\_object\_confidence[ ar\_object\_idx[ i ] ] syntax elements are not present. It is a requirement of bitstream conformance that the value of ar\_object\_confidence\_present\_flag shall be the same for all annotated\_regions( ) syntax structures within a CVS.

**ar\_object\_confidence\_length\_minus1** + 1 specifies the length, in bits, of the ar\_object\_confidence[ ar\_object\_idx[ i ] ] syntax elements. It is a requirement of bitstream conformance that the value of ar\_object\_confidence\_length\_minus1 shall be the same for all annotated\_regions( ) syntax structures within a CVS.

**ar\_object\_label\_language\_present\_flag** equal to 1 indicates that the ar\_object\_label\_language syntax element is present. ar\_object\_label\_language\_present\_flag equal to 0 indicates that the ar\_object\_label\_language syntax element is not present.

**ar\_bit\_equal\_to\_zero** shall be equal to zero.

**ar\_object\_label\_language** contains a language tag as specified by IETF RFC 5646 followed by a null termination byte equal to 0x00. The length of the ar\_object\_label\_language syntax element shall be less than or equal to 255 bytes, not including the null termination byte. When not present, the language of the label is unspecified.

**ar\_num\_label\_updates** indicates the total number of labels associated with the annotated regions that will be signalled. The value of ar\_num\_label\_updates shall be in the range of 0 to 255, inclusive.

**ar\_label\_idx**[ i ] indicates the index of the signalled label . The value of ar\_label\_idx[ i ]shall be in the range of 0 to 255, inclusive.

**ar\_label\_cancel\_flag** equal to 1 cancels the persistence scope of the ar\_label\_idx[ i ]-th label. ar\_label\_cancel\_flag equal to 0 indicates that the ar\_label\_idx[ i ]-th label will be assigned a signalled value.

**ar\_label**[ ar\_label\_idx[ i ] ] specifies the contents of the ar\_label\_idx[ i ] –th label. The length of the ar\_label[ ar\_label\_idx[ i ] ] syntax element shall be less than or equal to 255 bytes, not including the null termination byte.

**ar\_num\_object\_updates** indicates the number of object updates to be signalled. ar\_num\_object\_updates shall be in the range of 0 to 255, inclusive.

**ar\_object\_idx**[ i ] is the index of the object parameters to be signalled. ar\_object\_idx[ i ]shall be in the range of 0 to 255, inclusive.

**ar\_object\_cancel\_flag** equal to 1 cancels the persistence scope of the ar\_object\_idx[ i ]-th object. ar\_object\_cancel\_flag equal to 0 indicates that parameters associated with the ar\_object\_idx[ i ]-th object tracked object will be signalled.

**ar\_object\_label\_update\_flag** equal to 1 indicates that an object label will be signalled. ar\_object\_label\_update\_flag equal to 0 indicates that an object label will not be signalled.

**ar\_object\_label\_idx**[ ar\_object\_idx[ i ] ] indicates the index of the label corresponding to the ar\_object\_idx[ i ]-th object. When ar\_object\_label\_idx[ ar\_object\_idx[ i ] ] is not present, its value is inferred from a previous annotated regions SEI messages in output order in the same CVS, if any.

**ar\_bounding\_box\_update\_flag** equal to 1 indicates that object bounding box parameters will be signalled. ar\_bounding\_box\_update\_flag equal to 0 indicates that object bounding box parameters will not be signalled.

**ar\_bounding\_box\_cancel\_flag** equal to 1 cancels the persistence scope of the ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ]. ar\_partial\_object\_flag[ ar\_object\_idx[ i ] ], and ar\_object\_confidence[ ar\_object\_idx[ i ] ]. ar\_bounding\_box\_cancel\_flag equal to 0 indicates that ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ] ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] ar\_partial\_object\_flag[ ar\_object\_idx[ i ] ], and ar\_object\_confidence[ ar\_object\_idx[ i ] ] syntax elements will be signalled.

**ar\_bounding\_box\_top**[ ar\_object\_idx[ i ] ], **ar\_bounding\_box\_left**[ ar\_object\_idx[ i ] ], **ar\_bounding\_box\_width**[ ar\_object\_idx[ i ] ], and **ar\_bounding\_box\_height**[ ar\_object\_idx[ i ] ] specify the coordinates of the top-left corner and the width and height, respectively, of the bounding box of the ar\_object\_idx[ i ]-th object in the cropped decoded picture, relative to the conformance cropping window specified by the active SPS.

The value of ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ] shall be in the range of 0 to CroppedWidth / SubWidthC − 1, inclusive.

The value of ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ] shall be in the range of 0 to CroppedHeight / SubHeightC − 1, inclusive.

The value of ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ] shall be in the range of 0 to CroppedWidth / SubWidthC − ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], inclusive.

The value of ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] shall be in the range of 0 to CroppedHeight / SubHeightC − ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], inclusive.

The identified object rectangle contains the luma samples with horizontal picture coordinates from SubWidthC \* ( ConfWinLeftOffset + ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ] ) to SubWidthC \* ( ConfWinLeftOffset + ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ] + ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ] ) − 1, inclusive, and vertical picture coordinates from SubHeightC \* ( ConfWinTopOffset + ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ] ) to SubHeightC \* ( ConfWinTopOffset + ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ] + ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] ) − 1, inclusive.

The values of ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ] and ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] persist in output order within the CVS for each value of ar\_object\_idx[ i ]. When not present, the values of ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ] or ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] are inferred from a previous annotated regions SEI message in output order in the CVS, if any.

**ar\_partial\_object\_flag**[ ar\_object\_idx[ i ] ] equal to 1 indicates that the ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ] and ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] syntax elements represent the size and location of an object that is only partially visible within the cropped decoded picture. ar\_partial\_object\_flag[ ar\_object\_idx[ i ] ] equal to 0 indicates that the ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ], ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ] and ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] syntax elements represent the size and location of an object that may or may not be only partially visible within the cropped decoded picture. When not present, the value of ar\_partial\_object\_flag[ ar\_object\_idx[ i ] ] is inferred from a previous annotated regions SEI message in output order in the CVS, if any.

**ar\_object\_confidence**[ ar\_object\_idx[ i ] ] indicates the degree of confidence associated with the ar\_object\_idx[ i ]-th object, in units of 2−( ar\_object\_confidence\_length\_minus1 + 1 ), such that a higher value of ar\_object\_confidence[ ar\_object\_idx[ i ] ] indicates a higher degree of confidence. The length of the ar\_object\_confidence[ ar\_object\_idx[ i ] ] syntax element is ar\_object\_confidence\_length\_minus1 + 1 bits. When not present, the value of\_object\_confidence[ ar\_object\_idx[ i ] ] is inferred from a previous annotated regions SEI message in output order in the CVS, if any.

* 1. **Scalability dimension information SEI message**

8.19.1 Scalability dimension information SEI message syntax

|  |  |
| --- | --- |
| scalability\_dimension\_info( payloadSize ) { | **Descriptor** |
| **sdi\_max\_layers\_minus1** | u(6) |
| **sdi\_multiview\_info\_flag** | u(1) |
| **sdi\_auxiliary\_info\_flag** | u(1) |
| if( sdi\_multiview\_info\_flag  | |  sdi\_auxiliary\_info\_flag ) { |  |
| if( sdi\_multiview\_info\_flag ) |  |
| **sdi\_view\_id\_len** | u(4) |
| for( i = 0; i  <=  sdi\_max\_layers\_minus1; i++ ) { |  |
| if( sdi\_multiview\_info\_flag ) |  |
| **sdi\_view\_id\_val**[ i ] | u(v) |
| if( sdi\_auxiliary\_info\_flag ) |  |
| **sdi\_aux\_id**[ i ] | u(8) |
| } |  |
| } |  |
| } |  |

8.19.2 Scalability dimension information SEI message semantics

The scalability dimension information 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.

The bitstreamInScope is the sequence of AUs that consists, in decoding order, of the AU containing the current scalability dimension information 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 information SEI message.

[Ed. (JB): Should add some wording here and in VVC to describe how the i-th layer is defined.]

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

**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 information 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 information SEI message.

**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 information 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 information SEI message.

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

**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.

**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 X.

**Table X – 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 |

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.

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.

* 1. **Multiview acquisition information SEI message**

8.20.1 Multiview acquisition information SEI message syntax

|  |  |
| --- | --- |
| multiview\_acquisition\_info( payloadSize ) { | **Descriptor** |
| **intrinsic\_param\_flag** | u(1) |
| **extrinsic\_param\_flag** | u(1) |
| if( intrinsic\_param\_flag ) { |  |
| **intrinsic\_params\_equal\_flag** | u(1) |
| **prec\_focal\_length** | ue(v) |
| **prec\_principal\_point** | ue(v) |
| **prec\_skew\_factor** | ue(v) |
| for( i = 0; i <= intrinsic\_params\_equal\_flag ? 0 : numViewsMinus1; i++ ) { |  |
| **sign\_focal\_length\_x**[ i ] | u(1) |
| **exponent\_focal\_length\_x**[ i ] | u(6) |
| **mantissa\_focal\_length\_x**[ i ] | u(v) |
| **sign\_focal\_length\_y**[ i ] | u(1) |
| **exponent\_focal\_length\_y**[ i ] | u(6) |
| **mantissa\_focal\_length\_y**[ i ] | u(v) |
| **sign\_principal\_point\_x**[ i ] | u(1) |
| **exponent\_principal\_point\_x**[ i ] | u(6) |
| **mantissa\_principal\_point\_x**[ i ] | u(v) |
| **sign\_principal\_point\_y**[ i ] | u(1) |
| **exponent\_principal\_point\_y**[ i ] | u(6) |
| **mantissa\_principal\_point\_y**[ i ] | u(v) |
| **sign\_skew\_factor**[ i ] | u(1) |
| **exponent\_skew\_factor**[ i ] | u(6) |
| **mantissa\_skew\_factor**[ i ] | u(v) |
| } |  |
| } |  |
| if( extrinsic\_param\_flag ) { |  |
| **prec\_rotation\_param** | ue(v) |
| **prec\_translation\_param** | ue(v) |
| for( i = 0; i <= numViewsMinus1; i++ ) |  |
| for( j = 0; j < 3; j++ ) { /\* row \*/ |  |
| for( k = 0; k < 3; k++ ) { /\* column \*/ |  |
| **sign\_r**[ i ][ j ][ k ] | u(1) |
| **exponent\_r**[ i ][ j ][ k ] | u(6) |
| **mantissa\_r**[ i ][ j ][ k ] | u(v) |
| } |  |
| **sign\_t**[ i ][ j ] | u(1) |
| **exponent\_t**[ i ][ j ] | u(6) |
| **mantissa\_t**[ i ][ j ] | u(v) |
| } |  |
| } |  |
| } |  |

8.20.2 Multiview acquisition information SEI message semantics

The multiview acquisition information 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.

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.

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.

When the multiview acquisition information SEI message is included in a scalable nesting SEI message, the syntax elements bitstream\_subset\_flag, nesting\_op\_flag and all\_layers\_flag in the scalable nesting SEI message shall be equal to 0.

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 nesting\_num\_layers\_minus1.

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

In the semantics below, index i refers to the syntax elements and variables that apply to the layer with nuh\_layer\_id equal to nestingLayerIdList[ 0 ][ i ]. [Ed. (JB): How is nestingLayerIdList[ 0 ][ i ] determined? It exists in HEVC, but not in HEVC. For VVC, is this supposed to be a reference to NestingLayerId[ i ] ? ]

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 ] ) (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.

**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.

**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.

**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.

**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−prec\_focal\_length. The value of prec\_focal\_length shall be in the range of 0 to 31, inclusive.

**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−prec\_principal\_point. The value of prec\_principal\_point shall be in the range of 0 to 31, inclusive.

**prec\_skew\_factor** specifies the exponent of the maximum allowable truncation error for skew factor as given by 2−prec\_skew\_factor. The value of prec\_skew\_factor shall be in the range of 0 to 31, inclusive.

**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.

**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.

**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 Max( 0, prec\_focal\_length − 30 ).
* Otherwise (exponent\_focal\_length\_x[ i ] is in the range of 0 to 63, exclusive), the length is Max( 0, exponent\_focal\_length\_x[ i ] + 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.

**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.

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

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 Max( 0, prec\_focal\_length − 30 ).
* Otherwise (exponent\_focal\_length\_y[ i ] is in the range of 0 to 63, exclusive), the length is Max( 0, exponent\_focal\_length\_y[ i ] + prec\_focal\_length − 31 ).

**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.

**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.

**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 Max( 0, prec\_principal\_point − 30 ).
* Otherwise (exponent\_principal\_point\_x[ i ] is in the range of 0 to 63, exclusive), the length is Max( 0, exponent\_principal\_point\_x[ i ] + prec\_principal\_point − 31 ).

**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.

**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.

**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 Max( 0, prec\_principal\_point − 30 ).
* Otherwise (exponent\_principal\_point\_y[ i ] is in the range of 0 to 63, exclusive), the length is Max( 0, exponent\_principal\_point\_y[ i ] + prec\_principal\_point − 31 ).

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

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

**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.

**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 exponent\_skew\_factor[ i ] is equal to 0, the length is Max( 0, prec\_skew\_factor − 30 ).
* Otherwise (exponent\_skew\_factor[ i ] is in the range of 0 to 63, exclusive), the length is Max( 0, exponent\_skew\_factor[ i ] + prec\_skew\_factor − 31 ).

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

 (X)

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

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

**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.

**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.

**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 Max( 0, prec\_rotation\_param − 30 ).
* Otherwise (exponent\_r[ i ] is in the range of 0 to 63, exclusive), the length is Max( 0, exponent\_r[ i ] + prec\_rotation\_param − 31 ).

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

 (X)

**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.

**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.

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

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

 (X)

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 \* 2e − 31 \* ( 1 + n ÷ 2v).
* Otherwise (e is equal to 0), x is set equal to ( −1 )s \* 2−( 30 + v ) \* n.

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **x** | **s** | **e** | **n** |
| **focalLengthX**[ i ] | sign\_focal\_length\_x[ i ] | exponent\_focal\_length\_x[ i ] | mantissa\_focal\_length\_x[ i ] |
| **focalLengthY**[ i ] | sign\_focal\_length\_y[ i ] | exponent\_focal\_length\_y[ i ] | mantissa\_focal\_length\_y[ i ] |
| **principalPointX**[ i ] | sign\_principal\_point\_x[ i ] | exponent\_principal\_point\_x[ i ] | mantissa\_principal\_point\_x[ i ] |
| **principalPointY**[ i ] | sign\_principal\_point\_y[ i ] | exponent\_principal\_point\_y[ i ] | mantissa\_principal\_point\_y[ i ] |
| **skewFactor**[ i ] | sign\_skew\_factor[ i ] | exponent\_skew\_factor[ i ] | mantissa\_skew\_factor[ i ] |
| **rE**[ i ][ j ][ k ] | sign\_r[ i ][ j ][ k ] | exponent\_r[ i ][ j ][ k ] | mantissa\_r[ i ][ j ][ k ] |
| **tE**[ i ][ j ] | sign\_t[ i ][ j ] | exponent\_t[ i ][ j ] | mantissa\_t[ i ][ j ] |

* 1. **Depth representation information SEI message**

8.20.1 Depth representation information SEI message syntax

|  |  |
| --- | --- |
| depth\_representation\_info( payloadSize ) { | **Descriptor** |
| **z\_near\_flag** | u(1) |
| **z\_far\_flag** | u(1) |
| **d\_min\_flag** | u(1) |
| **d\_max\_flag** | u(1) |
| **depth\_representation\_type** | ue(v) |
| if( d\_min\_flag | | d\_max\_flag ) |  |
| **disparity\_ref\_view\_id** | 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 ) { |  |
| **depth\_nonlinear\_representation\_num\_minus1** | ue(v) |
| for( i = 1; i <= depth\_nonlinear\_representation\_num\_minus1 + 1; i++ ) |  |
| **depth\_nonlinear\_representation\_model**[ i ] |  |
| } |  |
| } |  |

##### 8.21.1 Depth representation information element syntax

|  |  |
| --- | --- |
| depth\_rep\_info\_element( OutSign, OutExp, OutMantissa, OutManLen ) { | **Descriptor** |
| **da\_sign\_flag** | u(1) |
| **da\_exponent** | u(7) |
| **da\_mantissa\_len\_minus1** | u(5) |
| **da\_mantissa** | u(v) |
| } |  |

8.21.1 Depth representation information SEI message semantics

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.

When present, the depth representation information SEI message shall be associated with one or more layers with AuxId 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.

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.

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

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.

**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.

**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.

**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.

**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.

**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.

The variable maxVal is set equal to ( 1  <<  ( 8 + bit\_depth\_luma\_minus8 ) ) − 1, where bit\_depth\_luma\_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 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). |
| 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.  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 ].  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 |

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

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.

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 \* 2e − 31 \* ( 1 + n ÷ 2v ).

– Otherwise (e is equal to 0), x is set equal to ( −1 )s \* 2−( 30 + v ) \* n.

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 |

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.

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

**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.

**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.

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.

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 ) )  
}

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.

##### 8.21.1.1 Depth representation information element semantics

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

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.

**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.

**da\_exponent** specifies the exponent of the floating-point value. The value of da\_exponent shall be in the range of 0 to 27 − 2, inclusive. The value 27 − 1 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 27 − 1 as indicating an unspecified value. The variable OutExp is set equal to da\_exponent.

**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.

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

* 1. **Alpha channel information SEI message**

8.20.1 Alpha channel information SEI message syntax

|  |  |
| --- | --- |
| alpha\_channel\_info( payloadSize ) { | **Descriptor** |
| **alpha\_channel\_cancel\_flag** | u(1) |
| if( !alpha\_channel\_cancel\_flag ) { |  |
| **alpha\_channel\_use\_idc** | u(3) |
| **alpha\_channel\_bit\_depth\_minus8** | u(3) |
| **alpha\_transparent\_value** | u(v) |
| **alpha\_opaque\_value** | u(v) |
| **alpha\_channel\_incr\_flag** | u(1) |
| **alpha\_channel\_clip\_flag** | u(1) |
| if( alpha\_channel\_clip\_flag ) |  |
| **alpha\_channel\_clip\_type\_flag** | u(1) |
| } |  |
| } |  |

8.20.1 Alpha channel information SEI message semantics

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.

For an auxiliary picture with nuh\_layer\_id equal to nuhLayerIdA and AuxId[ nuhLayerIdA ] equal to AUX\_ALPHA, an associated primary picture, if any, is a picture in the same access unit having AuxId[ 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.

When an access unit contains an auxiliary picture picA with nuh\_layer\_id equal to nuhLayerIdA and AuxId[ 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.

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.

**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.

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.

**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.

**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.

**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.

**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.

**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 Min( alpha\_opaque\_value, 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.

**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.

**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 ( alpha\_opaque\_value − 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 ( alpha\_opaque\_value − 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.

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.

* 1. **Extended DRAP indication SEI message**

8.20.1 Extended DRAP indication SEI message syntax

|  |  |
| --- | --- |
| extended\_drap\_indication( payloadSize ) { | **Descriptor** |
| **edrap\_rap\_id\_in\_clvs** | u(16) |
| **edrap\_leading\_pictures\_decodable\_flag** | u(1) |
| **edrap\_reserved\_zero\_12bits** | u(12) |
| **edrap\_num\_ref\_rap\_pics\_minus1** | u(3) |
| for( i = 0; i  <=  edrap\_num\_ref\_rap\_pics\_minus1; i++ ) |  |
| **edrap\_ref\_rap\_id**[ i ] | u(16) |
| } |  |

8.20.1 Extended DRAP indication SEI message semantics

The picture associated with an extended DRAP (EDRAP) indication SEI message is referred to as an EDRAP picture.

The presence of the EDRAP indication SEI message indicates that the constraints on picture order and picture referencing specified in this subclause apply. These constraints can enable a decoder to properly decode the EDRAP picture and the pictures that are in the same layer and follow it in both decoding order and output order without needing to decode any other pictures in the same layer except the list of pictures referenceablePictures, which consists of a list of IRAP or EDRAP pictures in decoding order that are within the same CLVS and identified by the edrap\_ref\_rap\_id[ i ] syntax elements.

The constraints indicated by the presence of the EDRAP indication SEI message, which shall all apply, are as follows:

– The EDRAP picture is a trailing picture.

– The EDRAP picture has a temporal sublayer identifier equal to 0.

– The EDRAP picture does not include any pictures in the same layer in the active entries of its reference picture lists except the referenceablePictures.

– Any picture that is in the same layer and follows the EDRAP picture in both decoding order and output order does not include, in the active entries of its reference picture lists, any picture that is in the same layer and precedes the EDRAP picture in decoding order or output order, with the exception of the referenceablePictures.

– When edrap\_leading\_pictures\_decodable\_flag is equal to 1, the following applies:

– Any picture that is in the same layer and follows the EDRAP picture in decoding order shall follow, in output order, any picture that is in the same layer and precedes the EDRAP picture in decoding order.

– Any picture that is in the same layer and follows the EDRAP picture in decoding order and precedes the EDRAP picture in output order does not include, in the active entries of its reference picture lists, any picture that is in the same layer and precedes the EDRAP picture in decoding order, with the exception of the referenceablePictures.

– Any picture in the list referenceablePictures does not include, in the active entries of its reference picture lists, any picture that is in the same layer and is not a picture at an earlier position in the list referenceablePictures.

NOTE – Consequenlty, the first picture in referenceablePictures, even when it is an EDRAP picture instead of an IRAP picture, does not include any picture from the same layer in the active entries of its reference picture lists.

**edrap\_rap\_id\_in\_clvs** specifies the RAP picture identifier, denoted as RapPicId, of the EDRAP picture.

Each IRAP or EDRAP picture is associated with a RapPicId value. The RapPicId value for an IRAP picture is inferred to be equal to 0. The RapPicId values for any two IRAP or EDRAP pictures within a CLVS shall be different.

**edrap\_reserved\_zero\_12bits** shall be equal to 0 in bitstreams conforming to this version of this Specification. Other values for edrap\_reserved\_zero\_12bits are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of edrap\_reserved\_zero\_12bits.

**edrap\_num\_ref\_rap\_pics\_minus1** plus 1 indicates the number of IRAP or EDRAP pictures that are within the same CLVS as the EDRAP picture and may be included in the active entries of the reference picture lists of the EDRAP picture.

**edrap\_ref\_rap\_id**[ i ] indicates RapPicId of the i-th IRAP or EDRAP picture that is within the same CLVS as the EDRAP picture and may be included in the active entries of the reference picture lists of the EDRAP picture.