|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  31st Meeting: San Diego, US, 13–20 Apr. 2018 | Document: JCTVC-AE1012-v1\_AH0025 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Annotated Regions SEI message for HEVC (Draft 2)** | | |
| *Status:* | Proposal to JCT-VC | | |
| *Purpose:* | Draft text for standardization | | |
| *Author(s) or Contact(s):* | Jill Boyce Gary J. Sullivan Ye-Kui Wang | Email: | [jill.boyce@intel.com](mailto:jill.boyce@intel.com) [garysull@microsoft.com](mailto:garysull@microsoft.com) [yekui.wang@huawei.com](mailto:yekui.wang@huawei.com) |
| *Source:* | Editors | | |

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

# Abstract

This document contains the draft text for changes to the High Efficiency Video Coding (HEVC) standard (Rec. ITU-T H.265 | ISO/IEC 23008-2) to specify the annotated regions SEI message, along with some corrections to the existing specification text.

**Changes to the specification text:**

*Replace D.2.1 with the following:*

**D.2.1 General SEI message syntax**

|  |  |
| --- | --- |
| sei\_payload( payloadType, payloadSize ) { | **Descriptor** |
| if( nal\_unit\_type  = =  PREFIX\_SEI\_NUT ) |  |
| if( payloadType  = =  0 ) |  |
| buffering\_period( payloadSize ) |  |
| else if( payloadType  = =  1 ) |  |
| pic\_timing( payloadSize ) |  |
| else if( payloadType  = =  2 ) |  |
| pan\_scan\_rect( payloadSize ) |  |
| else if( payloadType  = =  3 ) |  |
| filler\_payload( payloadSize ) |  |
| else if( payloadType  = =  4 ) |  |
| user\_data\_registered\_itu\_t\_t35( payloadSize ) |  |
| else if( payloadType  = =  5 ) |  |
| user\_data\_unregistered( payloadSize ) |  |
| else if( payloadType  = =  6 ) |  |
| recovery\_point( payloadSize ) |  |
| else if( payloadType  = =  9 ) |  |
| scene\_info( payloadSize ) |  |
| else if( payloadType  = =  15 ) |  |
| picture\_snapshot( payloadSize ) |  |
| else if( payloadType  = =  16 ) |  |
| progressive\_refinement\_segment\_start( payloadSize ) |  |
| else if( payloadType  = =  17 ) |  |
| progressive\_refinement\_segment\_end( payloadSize ) |  |
| else if( payloadType  = =  19 ) |  |
| film\_grain\_characteristics( payloadSize ) |  |
| else if( payloadType  = =  22 ) |  |
| post\_filter\_hint( payloadSize ) |  |
| else if( payloadType  = =  23 ) |  |
| tone\_mapping\_info( payloadSize ) |  |
| else if( payloadType  = =  45 ) |  |
| frame\_packing\_arrangement( payloadSize ) |  |
| else if( payloadType  = =  47 ) |  |
| display\_orientation( payloadSize ) |  |
| else if( payloadType  = =  56 ) |  |
| green\_metadata( payloadsize ) /\* specified in ISO/IEC 23001-11 \*/ |  |
| else if( payloadType  = =  128 ) |  |
| structure\_of\_pictures\_info( payloadSize ) |  |
| else if( payloadType  = =  129 ) |  |
| active\_parameter\_sets( payloadSize ) |  |
| else if( payloadType  = =  130 ) |  |
| decoding\_unit\_info( payloadSize ) |  |
| else if( payloadType  = =  131 ) |  |
| temporal\_sub\_layer\_zero\_index( payloadSize ) |  |
| else if( payloadType  = =  133 ) |  |
| scalable\_nesting( payloadSize ) |  |
| else if( payloadType  = =  134 ) |  |
| region\_refresh\_info( payloadSize ) |  |
| else if( payloadType  = =  135 ) |  |
| no\_display( payloadSize ) |  |
| else if( payloadType  = =  136 ) |  |
| time\_code( payloadSize ) |  |
| else if( payloadType  = =  137 ) |  |
| mastering\_display\_colour\_volume( payloadSize ) |  |
| else if( payloadType  = =  138 ) |  |
| segmented\_rect\_frame\_packing\_arrangement( payloadSize ) |  |
| else if( payloadType  = =  139 ) |  |
| temporal\_motion\_constrained\_tile\_sets( payloadSize ) |  |
| else if( payloadType  = =  140 ) |  |
| chroma\_resampling\_filter\_hint( payloadSize ) |  |
| else if( payloadType  = =  141 ) |  |
| knee\_function\_info( payloadSize ) |  |
| else if( payloadType  = =  142 ) |  |
| colour\_remapping\_info( payloadSize ) |  |
| else if( payloadType  = =  143 ) |  |
| deinterlaced\_field\_identification( payloadSize ) |  |
| else if( payloadType  = =  144 ) |  |
| content\_light\_level\_info( payloadSize ) |  |
| else if( payloadType  = =  145 ) |  |
| dependent\_rap\_indication( payloadSize ) |  |
| else if( payloadType  = =  146 ) |  |
| coded\_region\_completion( payloadSize ) |  |
| else if( payloadType  = =  147 ) |  |
| alternative\_transfer\_characteristics( payloadSize ) |  |
| else if( payloadType  = =  148 ) |  |
| ambient\_viewing\_environment( payloadSize ) |  |
| else if( payloadType  = =  149 ) |  |
| content\_colour\_volume( payloadSize ) |  |
| else if( payloadType  = =  150 ) |  |
| equirectangular\_projection( payloadSize ) |  |
| else if( payloadType  = =  151 ) |  |
| cubemap\_projection( payloadSize ) |  |
| else if( payloadType  = =  152 ) |  |
| fisheye\_video\_info( payloadSize ) |  |
| else if( payloadType  = =  154 ) |  |
| sphere\_rotation( payloadSize ) |  |
| else if( payloadType  = =  155 ) |  |
| regionwise\_packing( payloadSize ) |  |
| else if( payloadType  = =  156 ) |  |
| omni\_viewport( payloadSize ) |  |
| else if( payloadType  = =  157 ) |  |
| regional\_nesting( payloadSize ) |  |
| else if( payloadType  = =  158 ) |  |
| mcts\_extraction\_info\_sets( payloadSize ) |  |
| else if( payloadType  = =  159 ) |  |
| mcts\_extraction\_info\_nesting( payloadSize ) |  |
| else if( payloadType  = =  160 ) |  |
| layers\_not\_present( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  161 ) |  |
| inter\_layer\_constrained\_tile\_sets( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  162 ) |  |
| bsp\_nesting( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  163 ) |  |
| bsp\_initial\_arrival\_time( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  164 ) |  |
| sub\_bitstream\_property( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  165 ) |  |
| alpha\_channel\_info( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  166 ) |  |
| overlay\_info( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  167 ) |  |
| temporal\_mv\_prediction\_constraints( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  168 ) |  |
| frame\_field\_info( payloadSize ) /\* specified in Annex F \*/ |  |
| else if( payloadType  = =  176 ) |  |
| three\_dimensional\_reference\_displays\_info( payloadSize ) /\* specified in Annex G \*/ |  |
| else if( payloadType  = =  177 ) |  |
| depth\_representation\_info( payloadSize ) /\* specified in Annex G \*/ |  |
| else if( payloadType  = =  178 ) |  |
| multiview\_scene\_info( payloadSize ) /\* specified in Annex G \*/ |  |
| else if( payloadType  = =  179 ) |  |
| multiview\_acquisition\_info( payloadSize ) /\* specified in Annex G \*/ |  |
| else if( payloadType  = =  180 ) |  |
| multiview\_view\_position( payloadSize ) /\* specified in Annex G \*/ |  |
| else if( payloadType  = =  181 ) |  |
| alternative\_depth\_info( payloadSize ) /\* specified in Annex I \*/ |  |
| else if( payloadType  = =  200 ) |  |
| sei\_manifest( payloadSize ) |  |
| else if( payloadType  = =  201 ) |  |
| sei\_prefix\_indication( payloadSize ) |  |
| else if( payloadType  = =  202 ) |  |
| annotated\_regions( payloadSize ) |  |
| else |  |
| reserved\_sei\_message( payloadSize ) |  |
| else /\* nal\_unit\_type  = =  SUFFIX\_SEI\_NUT \*/ |  |
| if( payloadType  = =  3 ) |  |
| filler\_payload( payloadSize ) |  |
| else if( payloadType  = =  4 ) |  |
| user\_data\_registered\_itu\_t\_t35( payloadSize ) |  |
| else if( payloadType  = =  5 ) |  |
| user\_data\_unregistered( payloadSize ) |  |
| else if( payloadType  = =  17 ) |  |
| progressive\_refinement\_segment\_end( payloadSize ) |  |
| else if( payloadType  = =  22 ) |  |
| post\_filter\_hint( payloadSize ) |  |
| else if( payloadType  = =  132 ) |  |
| decoded\_picture\_hash( payloadSize ) |  |
| else if( payloadType  = =  146 ) |  |
| coded\_region\_completion( payloadSize ) |  |
| else |  |
| reserved\_sei\_message( payloadSize ) |  |
| if( more\_data\_in\_payload( ) ) { |  |
| if( payload\_extension\_present( ) ) |  |
| **reserved\_payload\_extension\_data** | u(v) |
| **payload\_bit\_equal\_to\_one** /\* equal to 1 \*/ | f(1) |
| while( !byte\_aligned( ) ) |  |
| **payload\_bit\_equal\_to\_zero** /\* equal to 0 \*/ | f(1) |
| } |  |
| } |  |

*Renumber clause D.2.47 (Reserved SEI message syntax) as D.2.48.*

*Add clause D.2.47, as follows:*

**D.2.47 Annotated regions SEI message syntax**

|  |  |
| --- | --- |
| annotated\_regions( payloadSize ) { | **Descriptor** |
| **ar\_cancel\_flag** | u(1) |
| **ar\_not\_optimized\_for\_viewing\_flag** | u(1) |
| **ar\_true\_motion\_flag** | u(1) |
| **ar\_occluded\_object\_flag** | u(1) |
| **ar\_partial\_object\_flag\_present\_flag** | u(1) |
| **ar\_object\_label\_present\_flag** | u(1) |
| **ar\_object\_confidence\_length** | u(4) |
| if( ar\_object\_label\_present\_flag ) { |  |
| **ar\_object\_label\_language\_present\_flag** | u(1) |
| if( ar\_object\_label\_language\_present\_flag ) { |  |
| while( !byte\_aligned( ) ) |  |
| **ar\_bit\_equal\_to\_zero** /\* equal to 0 \*/ | f(1) |
| **ar\_object\_label\_language** | st(v) |
| } |  |
| **ar\_num\_cancelled\_labels** | ue(v) |
| for( i = 0; i < ar\_num\_cancelled\_labels; i++ ) |  |
| **ar\_cancelled\_label\_idx**[ i ] | ue(v) |
| **ar\_num\_new\_labels** | ue(v) |
| for( i = 0; i < ar\_num\_new\_labels; i++ ) |  |
| **ar\_label\_idx**[ i ] | ue(v) |
| while( !byte\_aligned( ) ) |  |
| **ar\_bit\_equal\_to\_zero** /\* equal to 0 \*/ | f(1) |
| **ar\_label**[ ar\_label\_idx[ i ] ] | st(v) |
| } |  |
| } |  |
| **ar\_num\_cancelled\_objects** | ue(v) |
| for( i = 0; i < ar\_num\_cancelled\_objects; i++ ) |  |
| **ar\_cancelled\_object\_idx**[ i ] | ue(v) |
| **ar\_num\_objects\_minus1** | ue(v) |
| for( i = 0; i  <=  ar\_num\_objects\_minus1;i++ ) { |  |
| **ar\_object\_idx**[ i ] | ue(v) |
| **ar\_new\_object\_flag[** ar\_object\_idx[ i ] ] | u(1) |
| if( !ar\_new\_object\_flag[ ar\_object\_idx[ i ] ] ) |  |
| **ar\_bounding\_box\_update\_flag[** ar\_object\_idx[ i ] ] | u(1) |
| if( ar\_new\_object\_flag[ ar\_object\_idx[ i ]  &&  ar\_object\_label\_present\_flag ) |  |
| **ar\_object\_label\_idx**[ ar\_object\_idx[ i ] ] | ue(v) |
| if( ar\_partial\_object\_flag\_present\_flag ) |  |
| **ar\_partial\_object\_flag**[ ar\_object\_idx[ i ] ] | u(1) |
| if( ar\_bounding\_box\_update\_flag[ ar\_object\_idx[ i ] ]  | | ar\_new\_object\_flag[ ar\_object\_idx[ i ] ] ) { [Ed. (GJS): Suggest to insert byte alignment bits here or use ue(v) coding for the bounding box syntax elements, since this position is not byte aligned.] |  |
| **ar\_bounding\_box\_top[** ar\_object\_idx[ i ] ] | u(16) |
| **ar\_bounding\_box\_left**[ ar\_object\_idx[ i ] ] | u(16) |
| **ar\_bounding\_box\_width**[ ar\_object\_idx[ i ] ] | u(16) |
| **ar\_bounding\_box\_height**[ ar\_object\_idx[ i ] ] | u(16) |
| if( ar\_object\_confidence\_length > 0 )) |  |
| **ar\_object\_confidence**[ ar\_object\_idx[ i ] ] | u(v) |
| } |  |
| } |  |
| } |  |

*In D.3.1, replace the following paragraphs:*

The list SingleLayerSeiList is set to consist of the payloadType values 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, and 200 to 201, inclusive.

The list VclAssociatedSeiList is set to consist of the payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, and 200 to 201, inclusive.

The list PicUnitRepConSeiList is set to consist of the payloadType values 0, 1, 2, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 133, 135 to 152, inclusive, 154 to 159, inclusive, and 200 to 201, inclusive.

*with the following:*

The list SingleLayerSeiList is set to consist of the payloadType values 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, and 200 to 202, inclusive.

The list VclAssociatedSeiList is set to consist of the payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, and 200 to 202, inclusive.

The list PicUnitRepConSeiList is set to consist of the payloadType values 0, 1, 2, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 133, 135 to 152, inclusive, 154 to 159, inclusive, and 200 to 202, inclusive.

*In D.3.1, in Table D.1, append the following row to the end of the table:*

|  |  |
| --- | --- |
| Annotated regions | Specified by the syntax of the SEI message |

*Add clause D.3.47, as follows:*

**D.3.47 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.

**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.

Let picA be the current picture. Each region identified in the annotated regions SEI message persists for the current layer in output order until any of the following conditions are true:

– A new CLVS of the current layer begins.

– The bitstream ends.

– A picture picB in the current layer in an access unit containing an annotated regions SEI message that is applicable to the current layer is output for which PicOrderCnt( picB ) is greater than PicOrderCnt( picA ), where PicOrderCnt( picB ) and PicOrderCnt( picA ) are the PicOrderCntVal values of picB and picA, and the semantics of the annotated regions SEI message for PicB cancels the persistence of the region identified in the annotated regions SEI message for PicA.

**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 CLVS.

**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 CLVS.

**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. [Ed. (GJS): The terminology of the syntax element names and the semantics have switched from discussing “annotated regions” to discussing “annotated objects”. Is there a meaningful distinction, or is this just undesirable inconsistency?][Ed. (JB): There is not really a meaningful distinction. This SEI message is really tracking objects rather than regions, since the objects have labels and regions, and an object’s index can still be retained even when it isn’t within a current picture region. We could change the SEI message name to annotated objects, but perhaps better to not change its name.][Ed. (GJS): I think “region” is better for some purposes, since it avoids implying too much of a state of knowledge in the encoder, and since these are just rectangles. It is rectangles that are being annotated here. I think “object” makes sense when we need to talk about occlusion or confidence or something we are tracking. I think we don’t need either term in some cases. I changed the names of the bounding box syntax elements to use “bounding\_box” instead of “object”. I think we should not use the phrase “annotated objects”, although “objects in the annotated regions” may be OK. The objects should not be called annotated. Please review what I just did and see if you think it helps.]

**ar\_object\_confidence\_length** + 1 specifies the length, in bits, of the ar\_object\_confidence[ ar\_object\_idx[ i ] ] syntax elements. ar\_object\_confidence\_length 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\_length\_minus1 shall be the same for all annotated\_regions( ) syntax structures within a CLVS.

**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 and that the language of the label is unspecified.

**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. [Ed. (GJS): Is it OK for the language to change from picture to picture while the regions are persisting?] [Ed. (JB):I think it is OK to let change, to potentially allow different objects being tracked to have names in different languages. For a particular object being tracked, the syntax doesn’t allow re-labeling it.]

**ar\_num\_cancelled\_labels** indicates the number of cancelled labels associated with objects associated with the annotated regions. ar\_num\_cancelled\_labels shall be in the range of 0 to 255, inclusive.

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

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

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

**ar\_label**[ ar\_label\_idx[ i ] ] contains the label of the bounding box. 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\_cancelled\_objects** indicates the number of cancelled objects. ar\_num\_cancelled\_objects shall be in the range of 0 to 255, inclusive.

**ar\_cancelled\_object\_idx**[ i ] cancels the persistence of the object with index ar\_cancelled\_object\_idx[ i ] annotated object. The value of ar\_cancelled\_object\_idx[ i ]shall be in the range of 0 to 255, inclusive. [Ed. (GJS): Can the same index value be used again for a “new” object after an object with that index has been cancelled? Would that mean that the same object had reappeared or that we were just re-using the index value for something else since it was vacant?][Ed. (JB): The intention is the latter.][Ed. (GJS): Is there a way to indicate that some object has left the picture and then the same object has re-entered the picture?][Ed. (JB): No, there isn’t support for that.]

**ar\_num\_objects\_minus1** plus 1 indicates the total number of objects identified in the current cropped decoded picture. ar\_num\_objects\_minus1 shall be in the range of 0 to 255, inclusive.

**ar\_object\_idx**[ i ] is the index of the identified object in the list of objects identified in the current cropped decoded picture.ar\_object\_idx[ i ]shall be in the range of 0 to 255, inclusive.

**ar\_new\_object\_flag**[ ar\_object\_idx[ i] ] equal to 1 indicates that the corresponding object is not represented in an earlier annotated regions SEI message in output order within the CLVS. ar\_new\_object\_flag[ ar\_object\_idx[ i ] ] equal to 0 indicates that the corresponding object is represented in an earlier annotated regions SEI messages in output order within the CLVS.

**ar\_bounding\_box\_update\_flag**[ ar\_object\_idx[ i ] ] equal to 1 indicates that the bounding box of the corresponding object is not inferred from the bounding box specified in the previous annotated regions SEI message in output order in the CLVS with the same value of ar\_object\_idx[ i ]. ar\_bounding\_box\_update\_flag[ ar\_object\_idx[ i ] ] equal to 0 indicates that the bounding box of the corresponding object is inferred from the previous annotated regions SEI message in output order in the CLVS that contains the same value of ar\_object\_idx[ i ]. [Ed. (GJS): Is it OK to have ar\_object\_bounding\_box\_update\_flag[ ar\_object\_idx[ i ] ] equal to 0 when there was no previous previous annotated regions SEI message in output order in the CLVS that contains the same value of ar\_object\_idx[ i ]?][Ed. (JB): The syntax was designed to (hopefully) disallow that from happening.]

**ar\_object\_label\_idx**[ ar\_object\_idx[ i ] ] is the index of the label corresponding to the object. When ar\_object\_label\_idx[ ar\_object\_idx[ i ] ] is not present, its value is inferred from the value for the previous annotated regions SEI messages in output order in the same CLVS. If ar\_object\_label\_idx[ ar\_object\_idx[ i ] ] is not present in earlier annotated regions SEI messages in output order within the CLVS, its value is undefined.

**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 to be equal to 0.

**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 object with the index ar\_object\_idx[ i ] in the cropped decoded picture, relative to the conformance cropping window specified by the active SPS.

Let croppedWidth and croppedHeight be the width and height, respectively, of the cropped decoded picture in units of luma samples, as specified by Equations D-28 and D-29.

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 / SubHeightC − 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 \* ( conf\_win\_left\_offset + ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ] ) to SubWidthC \* ( conf\_win\_left\_offset + ar\_bounding\_box\_left[ ar\_object\_idx[ i ] ] + ar\_bounding\_box\_width[ ar\_object\_idx[ i ] ] ) − 1, inclusive, and vertical picture coordinates from SubHeightC \* ( conf\_win\_top\_offset + ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ] ) to SubWidthC \* ( conf\_win\_top\_offset + ar\_bounding\_box\_top[ ar\_object\_idx[ i ] ] + ar\_bounding\_box\_height[ ar\_object\_idx[ i ] ] ) − 1, inclusive. [Ed. (GJS): Would it be more consistent with other syntax usage to use left and right offsets and top and bottom offsets instead of a left offset and a width and a right offset and a height?][Ed. (JB): Either way would be OK, but we chose this since there is some value to having easy access to object size.][Ed. (GJS): I think that for the other places with similar needs, we originally did it that way to minimize syntax bits when the rectangle covered all or nearly all of the picture, which doesn’t apply here. On the other hand, this is using fixed-length coding, so there is no bit efficiency argument for how this is done.]

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 CLVS for each value of ar\_object\_idx[ i ]. If 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 ] ] were not present in earlier annotated regions SEI messages in output order within the CLVS with the same value of ar\_object\_idx[ i ], their values are undefined. [Ed. (GJS): Is it OK to infer a bounding box that is undefined?][Ed. (JB): It seemed easier to just say it was undefined that to try to figure out how to disallow the condition.]

**ar\_object\_confidence**[ ar\_object\_idx[ i ] ] indicates the degree of confidence associated with the object with index ar\_object\_idx[ i ], 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. The value of ar\_object\_confidence[ ar\_object\_idx[ i ] ] persists in output order within the CLVS for each value of ar\_object\_idx[ i ]. If ar\_object\_confidence[ ar\_object\_idx[ i ] ] is not present in an earlier annotated regions SEI messages in output order within the CLVS with the same value of ar\_object\_idx[ i ], its value is undefined. [Ed. (GJS): Is it OK to infer a confidence value that is undefined?] [Ed. (JB): It seemed easier to just say it was undefined that to try to figure out how to disallow the condition.]

*In F.14.3.1 (General SEI payload semantics), replace the following paragraphs:*

The list VclAssociatedSeiList is set to consist of the payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, 161, 165, 167, 168, and 200 to 201, inclusive.

The list PicUnitRepConSeiList is set to consist of the payloadType values 0, 1, 2, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 133, 135 to 152, inclusive, 154 to 168, inclusive, and 200 to 201, inclusive.

*with the following:*

The list VclAssociatedSeiList is set to consist of the payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, 161, 165, 167, 168, and 200 to 202, inclusive.

The list PicUnitRepConSeiList is set to consist of the payloadType values 0, 1, 2, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 133, 135 to 152, inclusive, 154 to 168, inclusive, and 200 to 202, inclusive.

*In G.14.3.1 (General SEI payload semantics), replace the following paragraphs:*

The list VclAssociatedSeiList is set to consist of payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, 161, 165, 167, 168, 177, 178, 179, and 200 to 201, inclusive.

The list PicUnitRepConSeiList is set to consist of payloadType values 0, 1, 2, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 133, 135 to 152, inclusive, 154 to 168, inclusive, 176 to 180, inclusive, and 200 to 201, inclusive.

*with the following:*

The list VclAssociatedSeiList is set to consist of payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, 161, 165, 167, 168, 177, 178, 179, and 200 to 202, inclusive.

The list PicUnitRepConSeiList is set to consist of payloadType values 0, 1, 2, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 133, 135 to 152, inclusive, 154 to 168, inclusive, 176 to 180, inclusive, and 200 to 202, inclusive.

*In I.14.3.1 (General SEI payload semantics), replace the following paragraphs:*

The list VclAssociatedSeiList is set to consist of payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, 161, 165, 167, 168, 177, 178, 179, and 200 to 201, inclusive.

The list PicUnitRepConSeiList is set to consist of payloadType values 0, 1, 2, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 133, 135 to 152, inclusive, 154 to 168, inclusive, 176 to 181, inclusive, and 200 to 201, inclusive.

*with the following:*

The list VclAssociatedSeiList is set to consist of payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, 161, 165, 167, 168, 177, 178, 179, and 200 to 202, inclusive.

The list PicUnitRepConSeiList is set to consist of payloadType values 0, 1, 2, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 133, 135 to 152, inclusive, 154 to 168, inclusive, 176 to 181, inclusive, and 200 to 202, inclusive.

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