 ISO/IEC JTC 1/SC 29/WG 5 N0018

**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 1 of ISO/IEC 23002-7 Amd.1 Additional SEI messages

**Status:** Approved

**Date of document:** 2020-10-16

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

**Expected action:** Info

**Action due date:** None

**No. of pages:** 6 (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**  20th Meeting, by teleconference, 7 – 16 Oct. 2020 | Document: JVET-T2017-v1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Additional SEI messages for VSEI (Draft 1)** | | |
| *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.264 | ISO/IEC 14496-10) to specify the annotated regions SEI message and for some technical corrections and editorial improvements.

**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.*

*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.

*Add clause 8.18 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.