**COMMITTEE DRAFT AMENDMENT****© ISO/IEC 2021 – All rights reserved****Text of ISO/IEC 23002-7:2021/CDAM 1** **63****Part 7: Versatile supplemental enhancement information for coded video bitstreams, AMENDMENT 1: Additional SEI messages****Information technology — MPEG video technologies****Élément introductif — Élément central — Partie 7: Titre de la partie****Information technology — MPEG video technologies — Part 7: Versatile supplemental enhancement information messages for coded video bitstreams, AMENDMENT 1: Additional SEI messages****E****2021-04-28****(30) Committee****ISO/IEC****ISO/IEC J****2021****1****Amendment****International Standard****202x****51****ISO/IEC 23002‑****ISO/IEC 23002‑7****ISO/IEC 23002-7:2021/CDAM 1****JISC****Coding of audio, picture, multimedia and hypermedia information****Information technology****5****29****1** **2****見出し 2****見出し 1****0****2****STD Version 2.1c2****30** **4** **ISO/IEC JTC 1/SC 29 /WG 5 N 51**

Date: **2021-04-28**

**Text of ISO/IEC 23002-7:2021/CDAM 1**

ISO/IEC JTC 1/SC 29/WG 5

Secretariat:  JISC

**Information technology — MPEG video technologies — Part 7: Versatile supplemental enhancement information messages for coded video bitstreams, AMENDMENT 1: Additional SEI messages**

*Élément introductif — Élément central — Partie 7: Titre de la partie*

|  |
| --- |
| **Warning**  This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.  Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation |

**Copyright notice**

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manger of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](https://www.iso.org/directives-and-policies.html)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](https://www.iso.org/iso-standards-and-patents.html)) or the IEC list of patent declarations received (see [http://patents.iec.ch](http://patents.iec.ch/)).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see [www.iso.org/iso/foreword.html](https://www.iso.org/foreword-supplementary-information.html).

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T. Technically aligned twin text is published as Rec. ITU-T H.274.

A list of all parts in the ISO/IEC 23002 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](https://www.iso.org/members.html).

Information technology — MPEG video technologies — Part 7: Versatile supplemental enhancement information messages for coded video bitstreams, AMENDMENT 1: Additional SEI messages

*Clause 2*

Add an additional normative reference as follows:

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

*Clause 4*

In Clause 4, add the following abbreviations:

|  |  |
| --- | --- |
| ACI | alpha channel information |
| CTI | colour transform information |
| DRI | depth representation information |
| MAI | multiview acquisition information |
| SDI | scalability dimension information |

*6.1*

In subclause 6.1, replace paragraphs 2 and 3 with the following:

Technical specifications that reference this document 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 strucure specified in Rec. ITU-T H.266 | ISO/IEC 23090-3. The design of the container 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 editions of this document, thus enabling this document to provide extensibility by directly appending additional syntax elements to the end of the vui\_parameters( ) syntax structure in future editions of this document. The syntax of the container of the vui\_parameters( ) syntax structure is outside the scope of this document.

Technical specifications that reference this document 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 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 editions of this document, thus enabling this document to provide extensibility by directly appending additional syntax elements to the end of the SEI message syntax structure in future editions of this document. 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 document.

*6.3*

In subclause 6.3, insert the following after the bullet item entry that starts with “se(v)”:

– 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 document when the current position in the bitstream is a byte-aligned position.

*8.1*

In subclause 8.1, replace Table 4 with the following:

**Table 4 – Persistence scope of SEI messages (informative)**

|  |  |
| --- | --- |
| **SEI message** | **Persistence scope** |
| Filler payload | The PU containing the SEI message |
| User data registered by Rec. ITU-T T.35 | Unspecified |
| User data unregistered | Unspecified |
| Film grain characteristics | Specified by the syntax of the SEI message |
| Frame packing arrangement | Specified by the syntax of the SEI message |
| Display orientation | Specified by the syntax of the SEI message |
| Referenced parameter sets | The CLVS containing the SEI message |
| Decoded picture hash | The PU containing the SEI message |
| Mastering display colour volume | The CLVS containing the SEI message |
| Colour transform information | Specified by the syntax of the SEI message |
| Content light level information | The CLVS containing the SEI message |
| DRAP indication | The picture associated with the SEI message |
| Alternative transfer characteristics | The CLVS containing the SEI message |
| Ambient viewing environment | The CLVS containing the SEI message |
| Content colour volume | Specified by the syntax of the SEI message |
| Equirectangular projection | Specified by the syntax of the SEI message |
| Generalized cubemap projection | Specified by the syntax of the SEI message |
| Sphere rotation | Specified by the syntax of the SEI message |
| Region-wise packing | Specified by the syntax of the SEI message |
| Omnidirectional viewport | Specified by the syntax of the SEI message |
| Alpha channel information | Specified by the syntax of the SEI message |
| Frame-field information | The PU containing the SEI message |
| Depth representation information | Specified by the semantics of the SEI message |
| Multiview acquisition information | The CVS containing the SEI message |
| Annotated regions | Specified by the syntax of the SEI message |
| Sample aspect ratio information | Specified by the syntax of the SEI message |
| Scalability dimension information | The CVS containing the SEI message |
| Extended DRAP indication | The picture associated with the SEI message |

*8.11.2*

Replace the contents of subclause 8.11.2 with the following:

**8.11.2 Dependent random access point indication SEI message semantics**

The presence of the DRAP 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 DRAP 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 associated IRAP picture of the DRAP picture.

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

– The DRAP picture is a trailing picture.

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

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

– Any picture that is in the same layer and follows the DRAP 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 DRAP picture in decoding order or output order, with the exception of the associated IRAP picture of the DRAP picture.

*8.16.1*

In subclause 8.16.1, in the row of the syntax table for ffi\_display\_elemental\_periods\_minus1, change “u(4)” to “u(8)”.

*8.18*

Renumber subclause 8.18 to 8.26 and renumber formulae 47 and 48 as 59 and 60.

Add subclauses 8.18 to 8.25 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\_minus1** | u(4) |
| for( i = 0; i  <=  sdi\_max\_layers\_minus1; i++ ) { |  |
| **sdi\_layer\_id**[ i ] | u(6) |
| if( sdi\_multiview\_info\_flag ) |  |
| **sdi\_view\_id\_val**[ i ] | u(v) |
| if( sdi\_auxiliary\_info\_flag ) |  |
| **sdi\_aux\_id**[ i ] | u(8) |
| if( sdi\_aux\_id[ i ] > 0 ) { |  |
| **sdi\_num\_associated\_primary\_layers\_minus1**[ i ] | u(6) |
| for( j = 0; j  <=  sdi\_num\_associated\_primary\_layers\_minus1[ i ]; j++ ) |  |
| **sdi\_associated\_primary\_layer\_idx**[ i ][ j ] | u(6) |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |

**8.19.2 Scalability dimension information SEI message semantics**

The scalability dimension information (SDI) SEI message provides the SDI for each layer in the current CVS, i.e., the CVS containing the SDI SEI message, such as 1) when there may be multiple views, 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, the auxiliary ID of each layer.

When an SDI SEI message is present in any AU of a CVS, an SDI SEI message shall be present for the first AU of the CVS. All SDI SEI messages in a CVS shall have the same content.

An SDI SEI message shall not be contained in a scalable nesting SEI message.

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

**sdi\_multiview\_info\_flag** equal to 1 indicates that the current CVS may have multiple views and the sdi\_view\_id\_val[ ] syntax elements are present in the SDI SEI message. sdi\_multiview\_info\_flag equal to 0 indicates that the current CVS does not have multiple views and the sdi\_view\_id\_val[ ] syntax elements are not present in the SDI SEI message.

**sdi\_auxiliary\_info\_flag** equal to 1 indicates that one or more layers in the current CVS may be auxiliary layers, which carry auxiliary information, and the sdi\_aux\_id[ ] syntax elements are present in the SDI SEI message. sdi\_auxiliary\_info\_flag equal to 0 indicates that the current CVS does not have an auxilary layer and the sdi\_aux\_id[ ] syntax elements are not present in the SDI SEI message.

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

**sdi\_layer\_id**[ i ] specifies the layer identifier of the i-th layer that may be present in the current CVS.

**sdi\_view\_id\_val**[ i ] specifies the view identifier of the i-th layer in the current CVS. The length of the sdi\_view\_id\_val[ i ] syntax element is sdi\_view\_id\_len\_minus1 + 1 bits.

The variable NumViews is derived as follows:

NumViews = 1  
if( sdi\_multiview\_info\_flag ) {  
 for( i = 1; i  <=  sdi\_max\_layers\_minus1; i++ ) {  
 newViewFlag = 1  
 for( j = 0; j < i; j++ ) (47)  
 if( sdi\_view\_id\_val[ i ]  = =  sdi\_view\_id\_val[ j ] )  
 newViewFlag = 0  
 if( newViewFlag )  
 NumViews++  
 }  
}

**sdi\_aux\_id**[ i ] equal to 0 indicates that the i-th layer in the current CVS does not contain auxiliary pictures. sdi\_aux\_id[ i ] greater than 0 indicates the type of auxiliary pictures in the i-th layer in the current CVS as specified in Table 15. When sdi\_auxiliary\_info\_flag is equal to 0, the value of sdi\_aux\_id[ i ] is inferred to be equal to 0.

**Table 15 – 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 document. 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 document, decoders shall allow values of sdi\_aux\_id[ i ] in the range of 0 to 255, inclusive.

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

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

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

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

* 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) |
| **num\_views\_minus1** | ue(v) |
| 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 : num\_views\_minus1; 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 <= num\_views\_minus1; 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 (MAI) SEI message specifies various parameters of the acquisition environment for the layers that may be present in the current CVS, i.e., the CVS containing the MAI SEI message. 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.

When an MAI SEI message is present in any AU of a CVS, an MAI SEI message shall be present for the first AU of the CVS. All MAI SEI messages in a CVS shall have the same content.

An MAI SEI message shall not be contained in a scalable nesting SEI message.

When a CVS does not contain an SDI SEI message, the CVS shall not contain an MAI SEI message.

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

Some of the views for which the MAI is included in an MAI SEI message may not be present in the current CVS.

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 ] ) (48)

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.

**num\_views\_minus1** plus 1 specifies the number of views for which the MAI is included in the MAI SEI message. The value of num\_views\_minus1 shall be equal to NumViews − 1.

**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

 (49)

**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:

 (50)

**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:

 (51)

The association between the camera parameter variables and corresponding syntax elements is specified by Table 16. Each component of the intrinsic and rotation matrices and the translation vector is obtained from the variables specified in Table 16 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 16 – 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**

**8.21.1 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 ] |  |
| } |  |
| } |  |

***8.21.1.1 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) |
| } |  |

**8.21.2 Depth representation information SEI message semantics**

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

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

– A bit depth for the samples of the luma component, denoted herein by BitDepthY.

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

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

When present, the DRI SEI message shall be associated with one or more layers that are indicated as depth auxiliary layers by an SDI SEI message. The following semantics apply separately to each nuh\_layer\_id targetLayerId among the nuh\_layer\_id values to which the DRI SEI message applies.

When present, the DRI 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.

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 DRI 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 17. In Table 17, 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  <<  BitDepthY ) − 1.

**Table 17 – 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 18 are derived from the respective variables in the s, e, n and v columns of Table 18 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 2 – The above specification is similar to that found in IEC 60559:1989.

**Table 18 – 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 3 – 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 ] (52)  
  
 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.

***8.21.2.1 Depth representation information element semantics***

The syntax structure specifies the value of an element in the DRI 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**

**8.22.1 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) |
| } |  |
| } |  |

**8.22.2 Alpha channel information SEI message semantics**

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

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

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

When an access unit contains an auxiliary picture picA in a layer that is indicated as an alpha auxiliary layer by an SDI SEI message, 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 ACI SEI message applies.

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

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

– A new CLVS of the current layer begins.

– The bitstream ends.

– A picture in the current layer in an AU associated with an ACI SEI message is output that follows the current picture in output order.

**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 plus 8 shall be equal to the bit depth 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**

**8.23.1 Extended DRAP indication SEI message syntax**

|  |  |
| --- | --- |
| extended\_drap\_indication( payloadSize ) { | **Descriptor** |
| **edrap\_rap\_id\_minus1** | 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) |
| } |  |

**8.23.2 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\_minus1** plus 1 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 EDRAP pictures associated with the same IRAP picture shall be different.

**edrap\_reserved\_zero\_12bits** shall be equal to 0 in bitstreams conforming to this version of this document. 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 RAP picture that may be included in the active entries of the reference picture lists of the EDRAP picture. The i-th RAP picture shall be either the IRAP picture associated with the current EDRAP picture or an EDRAP picture associated with the same IRAP picture as the current EDRAP picture.

* 1. **Display orientation SEI message**

**8.24.1 Display orientation SEI message syntax**

|  |  |
| --- | --- |
| display\_orientation( payloadSize ) { | **Descriptor** |
| **display\_orientation\_cancel\_flag** | u(1) |
| if( !display\_orientation\_cancel\_flag ) { |  |
| **display\_orientation\_persistence\_flag** | u(1) |
| **display\_orientation\_transform\_type** | u(3) |
| **display\_orientation\_reserved\_zero\_3bits** | u(3) |
| } |  |

**8.24.2 Display orientation SEI message semantics**

When the associated picture has PicOutputFlag equal to 1, the display orientation SEI message informs the decoder of a transformation that is recommended to be applied to the cropped decoded picture prior to display.

**display\_orientation\_cancel\_flag** equal to 1 indicates that the SEI message cancels the persistence of any previous display orientation SEI message in output order. display\_orientation\_cancel\_flag equal to 0 indicates that display orientation information follows.

**display\_orientation\_persistence\_flag** specifies the persistence of the display orientation SEI message for the current layer.

display\_orientation\_persistence\_flag equal to 0 specifies that the display orientation SEI message applies to the current decoded picture only.

display\_orientation\_persistence\_flag equal to 1 specifies that the display orientation SEI message applies to the current decoded picture and persists for all subsequent pictures of the current layer in output order until one or more of the following conditions are true:

– A new CLVS of the current layer begins.

– The bitstream ends.

– A picture in the current layer in an AU associated with a display orientation SEI message SEI message is output that follows the current picture in output order.

**display\_orientation\_transform\_type** specifies the rotation and mirroring to be applied to the picture. When display\_orientation\_transform\_type specifies both rotation and mirroring, rotation applies before mirroring. The values of display\_transform\_type are specified in Table 19.

**Table 19 – display\_orientation\_transform\_type values**

|  |  |
| --- | --- |
| **Value** | **Description** |
| 0 | no transform |
| 1 | mirroring horizontally |
| 2 | rotation by 180 degrees (anticlockwise) |
| 3 | rotation by 180 degrees (anticlockwise) before mirroring horizontally |
| 4 | rotation by 90 degrees (anticlockwise) before mirroring horizontally |
| 5 | rotation by 90 degrees (anticlockwise) |
| 6 | rotation by 270 degrees (anticlockwise) before mirroring horizontally |
| 7 | rotation by 270 degrees (anticlockwise) |

**display\_orientation\_reserved\_zero\_3bits** shall be equal to 0 in bitstreams conforming to this version of this document. Other values for display\_orientation\_reserved\_zero\_3bits are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of display\_\_orientation\_reserved\_zero\_3bits.

* 1. **Colour Transform Information SEI message**

**8.25.1 Colour Transform Information SEI message syntax**

|  |  |
| --- | --- |
| colour\_transform\_info( payloadSize ) { | **Descriptor** |
| **colour\_transform\_id** | ue(v) |
| **colour\_transform\_cancel\_flag** | u(1) |
| if( !colour\_transform\_cancel\_flag ) { |  |
| **colour\_transform\_persistence\_flag** | u(1) |
| **colour\_transform\_video\_signal\_info\_present\_flag** | u(1) |
| if( colour\_transform\_video\_signal\_info\_present\_flag ) { |  |
| **colour\_transform\_full\_range\_flag** | u(1) |
| **colour\_tranform\_primaries** | u(8) |
| **colour\_transform\_transfer\_function** | u(8) |
| **colour\_transform\_matrix\_coefficients** | u(8) |
| } |  |
| **colour\_transform\_bit\_depth\_minus8** | u(4) |
| **colour\_transform\_log2\_number\_of\_points\_per\_lut\_minus1** | u(3) |
| **colour\_transform\_cross\_component\_flag** | u(1) |
| if( colour\_transform \_cross\_component\_flag ) |  |
| **colour\_transform\_cross\_comp\_inferred\_flag** | u(1) |
| for( i = 0; i < colourTransformSize; i++ ) |  |
| **colour\_transf\_lut**[ 0 ][ i ] | u(v) |
| if( colour\_transform\_cross\_component\_flag  = =  0  | |  colour\_transform\_cross\_comp\_inferred\_flag  = =  0 ) { |  |
| **colour\_transform\_number\_chroma\_lut\_minus1** | u(1) |
| for( i = 0; i < colourTransformSize; i++ ) |  |
| **colour\_transf\_lut**[ 1 ][ i ] | u(v) |
| if( colour\_transform\_number\_chroma\_lut\_minus1 > 0 ) |  |
| for( i = 0; i < colourTransformSize; i++ ) |  |
| **colour\_transf\_lut**[ 2 ][ i ] | u(v) |
| } else |  |
| **colour\_transform\_chroma\_offset** | u(v) |
| } |  |
| } |  |

**8.25.2 Colour Transform Information SEI message semantics**

The Colour Transform Information (CTI) SEI message provides information to enable remapping of the reconstructed colour samples of the output pictures for purposes such as converting the output pictures to a representation that is more suitable for an alternative display. The colour transform model used in the CTI SEI message is composed of a first piece-wise linear function applied to the first colour component. Depending on the values of syntax elements colour\_transform\_cross\_component\_flag, colour\_transform\_cross\_comp\_inferred\_flag, and colour\_transform\_number\_chroma\_lut\_minus1, one or two additional piece-wise linear functions may be signalled for the second and third colour components.

When chroma\_format\_idc is equal to 0 (monochrome), the CTI SEI message shall not be present, although decoders shall allow such messages to be present and shall ignore any such CTI SEI messages that may be present.

**colour\_transform\_id** contains an identifying number that may be used to identify the purpose of the CTI. The value of colour\_transform\_id may be used (in a manner not specified in this document) to indicate that the input to the remapping process is the output of some conversion process that is not specified in this document, such as a conversion of the picture to some alternative colour representation (e.g., conversion from a YCbCr colour representation to a GBR colour representation). When more than one CTI SEI message is present with the same value of colour\_transform\_id, the content of these CTI SEI messages shall be the same. When CTI SEI messages are present that have more than one value of colour\_transform\_id, this may indicate that the remapping processes indicated by the different values of colour\_transform\_id are alternatives that are provided for different purposes or that a cascading of remapping processes is to be applied in a sequential order (an order that is not specified in this document). The value of colour\_transform\_id shall be in the range of 0 to 232 − 2, inclusive.

Values of colour\_transform\_id from 0 to 255 and from 512 to 231 − 1 may be used as determined by the application. Values of colour\_transform\_id from 256 to 511, inclusive, and from 231 to 232 − 2, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all CTI SEI messages containing a value of colour\_transform\_id in the range of 256 to 511, inclusive, or in the range of 231 to 232 − 2, inclusive, and bitstreams shall not contain such values.

NOTE – The colour\_transform\_id can be used to support different remapping processes that are suitable for different display scenarios. For example, different values of colour\_transform\_id may correspond to different remapped colour spaces supported by displays.

**colour\_transform\_cancel\_flag** equal to 1 indicates that the CTI SEI message cancels the persistence of any previous CTI SEI messages in output order that applies to the current layer. colour\_transform\_cancel\_flag equal to 0 indicates that CTI follows.

**colour\_transform\_persistence\_flag** specifies the persistence of the CTI SEI message for the current layer.

colour\_transform\_persistence\_flag equal to 0 specifies that the CTI SEI message applies to the current decoded picture only.

colour\_transform\_persistence\_flag equal to 1 specifies that the CTI SEI message applies to the current decoded picture and persists for all subsequent pictures of the current layer in output order until one or more of the following conditions are true:

– A new CLVS of the current layer begins.

– The bitstream ends.

– A picture in the current layer in an AU associated with a CTI SEI message is output that follows the current picture in output order.

**colour\_transform\_video\_signal\_info\_present\_flag** equal to 1 specifies that syntax elements colour\_transform\_full\_range\_flag, colour\_transform\_primaries, colour\_transform\_transfer\_function and colour\_transform\_matrix\_coefficients are present, colour\_transform\_video\_signal\_info\_present\_flag equal to 0 specifies that syntax elements colour\_transform\_full\_range\_flag, colour\_transform\_primaries, colour\_transform\_transfer\_function and colour\_transform\_matrix\_coefficients are not present.

**colour\_transform\_full\_range\_flag** has the same semantics as specified in clause 7.3 for the vui\_full\_range\_flag syntax element, except that colour\_transform\_full\_range\_flag identifies the colour space of the remapped reconstructed picture, rather than the colour space used for the CLVS. When not present, the value of colour\_transform\_full\_range\_flag is inferred to be equal to the value of vui\_full\_range\_flag.

**colour\_transform\_primaries** has the same semantics as specified in clause 7.3 for the vui\_colour\_primaries syntax element, except that colour\_transform\_primaries identifies the colour space of the remapped reconstructed picture, rather than the colour space used for the CLVS. When not present, the value of colour\_transform\_primaries is inferred to be equal to the value of vui\_colour\_primaries.

**colour\_transform\_transfer\_function** has the same semantics as specified in clause 7.3 for the vui\_transfer\_characteristics syntax element, except that colour\_transform\_transfer\_function identifies the colour space of the remapped reconstructed picture, rather than the colour space used for the CLVS. When not present, the value of colour\_transform\_transfer\_function is inferred to be equal to the value of vui\_transfer\_characteristics.

**colour\_transform\_matrix\_coefficients** has the same semantics as specified in clause 7.3 for the vui\_matrix\_coeffs syntax element, except that colour\_transform\_matrix\_coefficients identifies the colour space of the remapped reconstructed picture, rather than the colour space used for the CLVS. When not present, the value of colour\_transform\_matrix\_coefficients is inferred to be equal to the value of vui\_matrix\_coeffs.

**colour\_transform\_bit\_depth\_minus8** plus 8 specifies the bit depth of the colour components of the associated pictures for purposes of interpretation of the CTI SEI message. When any CTI SEI message is present with the value of colour\_transform\_bit\_depth plus 8 not equal to the bit depth of the decoded colour components, the SEI message refers to the hypothetical result of a conversion operation performed to convert the decoded colour component samples to the bit depth equal to colour\_transform\_input\_bit\_depth plus 8.

The value of colour\_transform\_bit\_depth plus 8 shall be in the range of 8 to 16, inclusive. Values of colour\_transform\_bit\_depth from in the range of 17 to 23, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all CTI SEI messages that contain a value of colour\_transform\_bit\_depth from in the range of 17 to 23, inclusive, and bitstreams shall not contain such values.

bitDepth is set equal to ( colour\_transform\_bit\_depth + 8 ).

**colour\_transform\_log2\_number\_of\_points\_per\_lut\_minus1** specifies the log2 of the number of pivot points in the piece-wise linear remapping functions minus 1.

log2numLutPoints is set equal to ( colour\_transform\_log2\_number\_of\_points\_per\_lut\_minus1 + 1 ).

numLutPoints is set equal to ( 1 << log2numLutPoints ).

colourTransformSize is set equal to ( numLutPoints + 1 ).

log2distX is set equal to ( bitDepth − log2numLutPoints ).

**colour\_transform\_cross\_component\_flag** equal to 1 indicates that the remapping of the second and third colour components is performed as cross-component remapping based on the first colour component. colour\_transform\_cross\_component\_flag equal to 0 indicates that intra-component remapping is applied to the second and third colour components.

maxIntraComp is set equal to ( 2 \* ( 1 − colour\_transform\_cross\_component\_flag ) ).

**colour\_transform\_cross\_comp\_inferred\_flag** equal to 1 indicates that the remapping piece-wise linear functions of the second and third colour components are inferred from the remapping piece-wise linear function of the first colour component. colour\_transform\_cross\_comp\_inferred\_flag equal to 0 indicates that the remapping piece-wise linear functions of the second and third colour components are signalled. When colour\_transform\_cross\_comp\_inferred\_flag is not present it is set equal to 0.

**colour\_transf\_lut**[ c ][ i ] specifies the piecewise linear remapping function of the colour component of index c. When colour\_transf\_lut[ 1 ][ i ] is present and colour\_transf\_lut[ 2 ][ i ] is not present, colour\_transf\_lut[ 2 ][ i ] is set equal to colour\_transf\_lut[ 1 ][ i ].

**colour\_transform\_number\_chroma\_lut\_minus1** equal to 1 indicates that colour\_transf\_lut[ 2 ][ i ] is present in the CTI SEI message. colour\_transform\_number\_chroma\_lut\_minus1 equal to 0 indicates that colour\_transf\_lut[ 2 ][ i ] is not present in the CTI SEI message. When colour\_transform\_number\_chroma\_lut\_minus1 is not present it is set equal to 0.

**colour\_transform\_chroma\_offset** specifies the CTI chroma offset. When not present, colour\_transform\_chroma\_offset is inferred to be equal to 0.

The remapping process of the input picture components rec[ c ], with width and height equal to picWidth[ c ] and picHeight[ c ], respectively, to the output remapped picture components map[ c ], for c = 0..2, is performed as follows.

The array pivotPointX is derived as follows.

– For j = 0..( numLutPoints − 1 ), pivotPointX[ j ] is set equal to ( j << log2distX ).

For c = 0..maxIntraComp, the arrays pivotPointY[ c ] and slope[ c ] are derived as follows:

– pivotPointY[ c ][ 0 ] is set equal to colour\_transf\_lut[ c ][ 0 ]

– For j = 1..( numLutPoints − 1 ), pivotPointY[ c ][ j ] is derived as follows:

pivotPointY[ c ][ j ] = pivotPointY[ c ][ j − 1 ] + colour\_transf\_lut[ c ][ j ] (53)

– For j = 0..( numLutPoints − 1 ), slope[ c ][ j ] is derived as follows:

slope[ c ][ j ] = ( ( colour\_transf\_lut[ c ][ j + 1 ] << 11 ) + ( 1 << ( log2distX − 1 ) ) ) >>   
 log2distX (54)

When colour\_transform\_cross\_component\_flag is equal to 1, the arrays ccPivotPointY[ c ] and ccSlope[ c ] are derived as follows, for c = 1..2:

– If colour\_transform\_cross\_comp\_inferred\_flag is equal to 0, ccPivotPointY[ c ] is derived as follows:

– For j = 0..numLutPoints, ccPivotPointY[ c ][ j ] is set equal to ( colour\_transf\_lut[ c ][ j ]  <<  ( 11 − log2distX ) ).

– Otherwise (colour\_transform\_cross\_comp\_inferred\_flag is equal to 1), ccPivotPointY[ c ] is derived as follows:

– For j = 0..( numLutPoints − 1 ), tmpPivotPt[ j ] is derived as follows:

– If colour\_transf\_lut[ 0 ][ j + 1 ] is equal to 0, tmpPivotPt[ j ] is set equal to ( 1 << 11 ).

– Otherwise, tmpPivotPt[ j ] is derived as follows:

tmpPivotPt[ j ] = ( colour\_transf\_lut[ 0 ][ j + 1 ] + colour\_transform\_chroma\_offset ) << (55)  
 ( 11 – log2distX )

– The array ccPivotPointY[ c ] is derived as follows:

– For j = 1..( numLutPoints − 1 ), ccPivotPointY[ c ][ j ] is derived as follows:

ccPivotPointY[ c ][ j ] = ( tmpPivotPt[ j ] + tmpPivotPt[ j − 1 ] + 1 ) / 2 (56)

– ccPivotPointY[ c ][ 0 ] is set equal to tmpPivotPt[ 0 ].

– ccPivotPointY[ c ][ numLutPoints ] is set equal to tmpPivotPt[ numLutPoints − 1 ].

– For j = 0..( numLutPoints − 1 ), the value of ccSlope[ c ][ j ] is set equal to ( ccPivotPointY[ c ][ j + 1 ] − ccPivotPointY[ c ][ j ] ).

For c = 0..maxIntraComp, the intra-component remapping process of the input samples picture rec[ c ] into the remapped samples picture map[ c ] is performed as follows.

– for i = 0..picWidth[ c ] − 1, j = 0..picHeight[ c ] − 1, the following applies:

idx = rec[ c ][ i ][ j ] >> log2distX  
map[ c ][ i ][ j ] = Clip3( 0, ( 1 << bitDepth ) − 1, pivotPointY[ c ][ idx ] + (57)  
 ( ( slope[ c ][ idx ] \* ( rec[ i ][ j ] − pivotPointX[ idx ] ) + ( 1 << 10 ) ) >> 11 ) )

When colour\_transform\_cross\_component\_flag is equal to 1, for c = 1..2, the cross-component remapping process of the input samples picture rec[ c ] into the remapped samples picture map[ c ] is performed as follows:

– offset is set equal to ( 1 << ( bitDepth − 1 ) ).

– subWc and subHc are set equal to ( picWidth[ 0 ] / picWidth[ c ] ) and ( picHeight[ 0 ] / picHeight[ c ] ), respectively.

– For i = 0..picWidth[ c ] − 1, j = 0..picHeight[ c ] − 1, the following applies:

coloc = rec[ 0 ][ i \* SubWc ][ j \* SubHc ]  
idx = coloc >> log2distX (58)  
scale = ccPivotPointY[ c ][ idx ] + ( ( ccSlope[ c ][ idx ] \* ( coloc − pivotPointX[ idx ] ) ) >> log2distX )  
map[ c ][ i ][ j ] = Clip3( 0, ( 1 << bitDepth ) − 1,  
 ( ( offset << 11 ) + scale \* ( rec[ c ][ i ][ j ] − offset ) + ( 1 << 10 ) ) >> 11 )

*9.1 and 9.2*

In subclauses 9.1 and 9.2, renumber Tables 15 to 17 as Tables 20 to 22.

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