|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  39th Meeting: by teleconference, 18–24 April 2020 | Document: JCTVC-AM1004-v3 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Errata report items for HEVC, AVC, Video CICP and Codepoint Usage Technical Report** | | |
| *Status:* | Output document approved by JCT-VC | | |
| *Purpose:* | Errata report | | |
| *Author(s) or Contact(s):* | **Gary Sullivan**  **Yasser Syed**  **Ye-Kui Wang** | Tel: Email: | [garysull@miscrosoft.com](mailto:garysull@miscrosoft.com)  [yasser\_syed@comcast.com](mailto:yasser_syed@comcast.com)  [yekui.wang@bytedance.com](mailto:yekui.wang@bytedance.com) |
| *Source:* | Editors | | |

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

# Abstract

This document contains a list of reported errata items for HEVC, AVC, Video CICP, and the TR on usage of video signal type code points, for tracking purposes. Some of the items have been confirmed by the JCT-VC and have been agreed to require fixing, while some other items have not yet been confirmed.

# General aspects and issues affecting multiple standards

## General and minor matters

Please note that the alignment between the text published by ITU-T and that published by ISO/IEC should also be checked. Minor editorial issues and aspects that are highlighted for potential further checking include the following:

* (for HEVC only, and for the ITU-T text only; for the ISO/IEC text, this has been corrected in w18277\_FDIS\_23008-2\_4thEd\_plusAmd1\_w18881) In the semantics of matrix\_coeffs, there seems to be a problem in the HEVC spec (the AVC text is OK), saying "one or more" instead of "both" regarding the following constraint:

matrix\_coeffs shall not be equal to 0 unless both of the following conditions are true:

– BitDepthC is equal to BitDepthY.

– chroma\_format\_idc is equal to 3 (the 4:4:4 chroma format).

* Updating the reference to Rec. ITU-R BT.2100 (the current version being BT.2100-2) and the associated equations for the ICTCP matrix coefficients interpretation for HLG (esp. check Video CICP; this is correct in Rec. ITU-T H.265 2019-06).
* (for Video CICP only, in the ISO/IEC version only) Logical indentation nesting in the ISO/IEC version of the Video CICP colour interpretation was inadvertently removed in the publication process.
* (for Video CICP only) The range of values for the extended aspect ratio indicator in Video CICP is not clearly specified and may implicitly be interpreted as inadequate to cover the range of values expressed in the video coding standards.

## sYCC colour indicator interpretation

### Status

This item has been confirmed by the JCT-VC and resolved for the ITU-T text, but has not yet been resolved for the ISO/IEC text. It affects multiple standards: HEVC, AVC, and Video CICP (and JPEG XR). For background, see [JCTVC-AJ0023](http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=10941).

### Description of the issue

This issue relates to the transfer characteristics and matrix coefficients indicators for the sYCC colour representation specified in IEC 61966-2-1. The the semantics of transfer characteristics (Table E.4 of HEVC), and matrix coefficients (Table E.5 of HEVC) need correction to address the issue.

The issue is a bit complicated because the same transfer characteristics indicator value is used for both sRGB and sYCC, but IEC 61966-2-1 actually indicates that the transfer characteristics function should be somewhat different for the two cases. In the sRGB case, the range of the input value is constrained to be from 0 to 1, but in the sYCC case, this constraint should not apply.

The agreed correction for this aspect is to condition the interpretation of the transfer\_characteristics syntax element for the value 13 on the value of the matrix\_coeffs syntax element (since that value would differ between sYCC and sRGB).

Additionally, the the informative remark relating to the matrix coefficients indicator value for IEC 61966-2-1 sYCC should be changed to indicate that sYCC should be indicated with the matrix coefficients indicator equal to 5 (as for Rec. ITU R BT.601) rather than 1 (as for Rec. ITU-R BT.709).

### Description of the proposed fix

*In E.3.1 (VUI semantics), in Table E.4 (Transfer characteristics interpretation using the transfer\_characteristics syntax element) replace the row for the value 13 with:*

|  |  |  |
| --- | --- | --- |
| 13 | – If matrix\_coeffs is equal to 0           V = *α* \* Lc( 1 ÷ 2.4 ) − ( *α* − 1 ) for 1 >= Lc >= *β*           V = 12.92 \* Lc for *β* > Lc >= 0  – Otherwise           V = *α* \* Lc( 1 ÷ 2.4 ) − ( *α* − 1 ) for Lc >= *β*           V = 12.92 \* Lc for *β* > Lc > −*β*           V = − *α* \* ( −Lc )( 1 ÷ 2.4 ) + ( *α* − 1 ) for −*β* >= Lc | IEC 61966-2-1 sRGB (with matrix\_coeffs equal to 0)  IEC 61966-2-1 sYCC (with matrix\_coeffs equal to 5) |

*In E.3.1 (VUI semantics), after Equation E-3, replace the next paragraph and associated bullet points with:*

In this case, the range of E′R, E′G, and E′B is specified as follows:

– If transfer\_characteristics is equal to 11 or 12, or transfer\_characteristics is equal to 13 and matrix\_coeffs is not equal to 0, E′R, E′G, and E′B are real numbers with values that have a larger range than the range of 0 to 1, inclusive, and their range is not specified in this Specification.

– Otherwise, E′R, E′G and E′B are real numbers in the range of 0 to 1.

*In E.3.1 (VUI semantics), in Table E.5 (Matrix coefficients interpretation using the matrix\_coeffs syntax element), move “*IEC 61966-2-1 sYCC*” from the row for the value 1 to the row for the value 5.*

# Reported errata items for HEVC

See also section 1.

## Publication status background

Rec. ITU-T H.265

* (02/18, Edition 5) Approved 2018-02-13, published 2018-05-11
* (06/19, Edition 6) Approved 2019-06-29, published 2019-09-23
* (10/19, Edition 7) Approved 2019-11-29, published 2020-01-10

ISO/IEC 23008-2

* ISO/IEC 23008-2:2017 (Edition 3); published 2017-10
* ISO/IEC 23008-2:2017/Amd 1:2018 (Additional colour representation code point); published 2018-03
* ISO/IEC 23008-2:2017/Amd 2:2018 (Main 10 still picture profile); published 2018-03
* ISO/IEC 23008-2:2017/Amd 3:2018 (Additional supplemental enhancement information); published 2018-07
* Work items merged for FDIS ballot as publication office action; Stage 50.00 Final text received or FDIS registered for formal approval as of 2020-03-19
  + ISO/IEC DIS 23008-2:201x (Edition 4); DIS ballot closed 2018-10-06; stage 40.99, Full report circulated: DIS approved for registration as FDIS 2019-02-18; Pending FDIS ballot based on WG 11 N 18277
  + ISO/IEC 23008-2:201x (Edition 4)/DAmd 1:201x (Additional supplemental enhancement information); DAM ballot started 2019-07-10, closed 2019-10-02

## On the general decoding process

### Status

This item has been confirmed by the JCT-VC, and the proposed fix has been agreed in spirit. For background, see [JCTVC-AI0022](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10923) and [JCTVC-AJ0021](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10939). However, the specific text for the fix is not yet available and hence this has not yet been resolved.

### Description of the issue

The general decoding process is specified in a CVS-by-CVS manner, because the sub-bitstream extraction process relies on the SPS syntax element sps\_max\_sub\_layers\_minus1, while the active SPS can change across CVSs.

For each CVS of the input bitstream, the sub-bitstream extraction process is applied first with the output being BitstreamToDecode. Therefore, BitstreamToDecode contains one CVS. After that, the decoding process for a coded picture specified in clause 8.1.3 is repeatedly invoked for each coded picture in BitstreamToDecode.

The following is stated in clause 8.1.3:

*The decoding process for the current picture takes as inputs the syntax elements and upper-case variables from clause 7. When interpreting the semantics of each syntax element in each NAL unit, the term "the bitstream" (or part thereof, e.g., a CVS of the bitstream) refers to BitstreamToDecode (or part thereof).*

From the yellow-highlighted wording, it is clear that the term "the bitstream" mentioned above was intended to mean something that can contain more than one CVS.

Furthermore, there are many places related to "*interpreting the semantics of each syntax element in each NAL unit*" that have the wording "the first picture in the bitstream", particularly when used to determine whether a CRA picture starts a CVS. In my understanding, in those places, the intent was that the term "the bitstream" mentioned above was intended to mean something that can contain more than one CVS.

Lastly, at the beginning of the general decoding process for the input bitstream, strictly speaking, determining the CVSs in the input bitstream would not be clear before determination of the variable NoRaslOutputFlag for each IRAP picture, which currently is only part of the decoding process of a CVS.

### Description of the proposed fix

It is proposed to change the description of the general decoding process as follows:

1. Firstly the CVSs in the input bitstream are identified.
2. Sub-bitstream extraction is applied for each CVS.
3. The extracted results of all the CVSs are concatenated to BitstreamToDecode.
4. BitstreamToDecode is decoded CVS by CVS, and each CVS picture by picture.

## On HRD for splicing

### Status

This item has been confirmed by the JCT-VC and resolved for the ITU-T text, but has not yet been resolved for the ISO/IEC text. For background, see [JCTVC-AK0027](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10967).

### Description of the issue

It is reported that the current HEVC specification determines, when concatenationFlag is set to 1, that the AuNominalRemovalTime for the splicing point under some circumstances (non-seamless splicing) may be greater than AuNominalRemovalTime[ prevNonDiscardablePic ] + ( auCpbRemovalDelayDeltaMinus1 + 1 ).

The text in HEVC specifies the following:

if( !concatenationFlag ) {  
 baseTime = AuNominalRemovalTime[ firstPicInPrevBuffPeriod ]  
 tmpCpbRemovalDelay = AuCpbRemovalDelayVal  
} else {  
 baseTime = **AuNominalRemovalTime[ prevNonDiscardablePic ]**  
 tmpCpbRemovalDelay =  
 Max( ( auCpbRemovalDelayDeltaMinus1 + 1 ), (C‑10)  
 Ceil( ( **InitCpbRemovalDelay[ SchedSelIdx ] ÷ 90000 +  
 AuFinalArrivalTime[ n − 1 ] − AuNominalRemovalTime[ n − 1 ]** ) ÷ ClockTick ) )  
}  
AuNominalRemovalTime[ n ] = baseTime + ClockTick \* ( tmpCpbRemovalDelay − CpbDelayOffset )

In the following figure, a splicing operation is shown that reportedly does not allow for seamless splicing since the InitCpbRemovalDelay[ SchedSelIdx ] imposes a later removal time after splicing than equidistant CPB removal times as present in seamless playout.



If instead of deriving the removal time of AU 6 from the AU n-1 (i.e. AU 5) as indicated above in the figure, we use the last non-discardable picture (AU 4), this would reportedly result in the following:

taf6= trm4 + >= trm6 (CPB underflow resulting in non-seamless splicing)

trm6= taf5+InitCpbRemovalDelay

=taf6 – trm4= Size / Bitrate – ( trm4 – taf5 )

trm4 + Size / Bitrate – ( trm4 – taf5 ) >= taf5 + InitCpbRemovalDelay

Size / Bitrate – ( trm4 – taf5 ) >= ( taf5 – trm4 )+ InitCpbRemovalDelay

taf5 + Size / Bitrate = **taf6 >= trm4 + ( taf5 – trm4 ) + InitCpbRemovalDelay**

In the derived formula, the part marked in yellow highlight and boldface in Equation C-10 above is reportedly not correct.

### Description of the proposed fix

The proposed fix is to change Equation C-10 to the following:

if( !concatenationFlag ) {  
 baseTime = AuNominalRemovalTime[ firstPicInPrevBuffPeriod ]  
 tmpCpbRemovalDelay = AuCpbRemovalDelayVal  
} else {  
 baseTime1 = AuNominalRemovalTime[ prevNonDiscardablePic ]  
 tmpCpbRemovalDelay1 = ( auCpbRemovalDelayDeltaMinus1 + 1 )  
 baseTime2 = AuNominalRemovalTime[ n − 1 ]  
 tmpCpbRemovalDelay2 =   
 Ceil( ( InitCpbRemovalDelay[ SchedSelIdx ] ÷ 90000 +  
 AuFinalArrivalTime[ n − 1 ] − AuNominalRemovalTime[ n − 1 ] ) ÷ ClockTick ) (C‑10)  
 if( baseTime1 + ClockTick \* tmpCpbRemovalDelay1 <   
 baseTime2 + ClockTick \* tmpCpbRemovalDelay2 ) {  
 baseTime = baseTime2  
 tmpCpbRemovalDelay = tmpCpbRemovalDelay2  
 } else {  
 baseTime = baseTime1  
 tmpCpbRemovalDelay = tmpCpbRemovalDelay1  
 }  
}  
AuNominalRemovalTime[ n ] = baseTime + ClockTick \* ( tmpCpbRemovalDelay − CpbDelayOffset )

## On filtering process for chroma block edges

### Status

This item has been confirmed by the JCT-VC and resolved for the ITU-T text, but has not yet been resolved for the ISO/IEC text.

### Bug fix

*In 8.7.2.5.5 (Filtering process for chroma block edges), change the following:*

If ChromaArrayType is equal to 1, the variable QpC is determined as specified in Table ‎8‑10 based on the index qPi derived as follows:

qPi = ( ( QpQ + QpP + 1 )  >>  1 ) + cQpPicOffset (‎8‑384)

Otherwise (ChromaArrayType is greater than 1), the variable QpC is set equal to Min( qPi, 51 ).

*to*

The index qPi derived as follows:

qPi = ( ( QpQ + QpP + 1 )  >>  1 ) + cQpPicOffset (8‑384)

The variable QpC is derived as follows:

– If ChromaArrayType is equal to 1, the variable QpC is determined based on qPi as specified in Table 8‑10.

– Otherwise (ChromaArrayType is greater than 1), the variable QpC is set equal to Min( qPi, 51 ).

## Absence of “persistence flag” for the annotated region SEI message

In all SEI messages that have a "cancel flag", there is also a corresponding "persistence flag", except in the case of the newly adopted annotated regions SEI message. Without this flag, it is not possible to indicate that the annotated regions information applies only to the current picture. Due to the extensibility provision that is built into the SEI message syntax, it would be possible to append a persistence flag at the end of the syntax to correct this likely oversight. The presence of the extra flag could be optional (conditioned on whether there is more data in the SEI message), so that encoders built for the original syntax would still be conforming. When the flag is not present, it would be inferred to be equal to 1.

## Miscellaneous issues for the published ITU-T H.265 version 7 (potentially also applicable to the ISO/IEC 23008-2 text)

1. In clause 5.10, replace "a mixture of lower case and upper case letter" with " a mixture of lower case and upper case letters" (missing 's' in the last word).
2. In clause 6.3.3, replace "The following divisions of processing elements of this Specification form spatial or component-wise partitioning:" with "The following divisions of processing elements of this Specification form spatial or component-wise partitionings:" (missing 's' in the last word)? Note that in the ISO version there is the 's'. Which one is (more) correct?
3. In clause 7.2, definition of next\_bits( n ), replace "Provides a look at the next n bits in the bitstream with n being its argument." with "It provides a look at the next n bits in the bitstream with n being its argument."
4. In clause 7.3.3, in the syntax condition for general\_max\_14bit\_constraint\_flag, replace the "Enter" with "shift+Enter".
5. In clause 7.3.3, in the syntax condition for sub\_layer\_max\_14bit\_constraint\_flag[ i ], replace two instances of "Enter" with "shift+Enter".
6. In clause 7.3.3, consider ticket #1504 Small typos in profile\_tier\_level syntax in tabular form: Aspect 1: Extra ')' is still present in H.265 (11/19). Aspect 2: sub\_layer\_profile\_compatibility\_flag missing “[ i ]” issue, confirmed fixed in H.265 (11/19).
7. In clause 7.4.4, remove the instance of "and " highlighted below:

When profilePresentFlag is equal to 1, general\_profile\_idc is not equal to 9 or 11 and is not in the range of 1 to 5, inclusive, and general\_profile\_compatibility\_flag[ 9 ] is not equal to 1, general\_profile\_compatibility\_flag[ 11 ] is not equal to 1, and general\_profile\_compatibility\_flag[ j ] is not equal to 1 for any value of j in the range of 1 to 5, inclusive, the value of general\_inbld\_flag is inferred to be equal to 0.

1. In clause 8.4.4.27, replace the following equations:

xL = palette\_transpose\_flag ? x \* nSubHeight : x \* nSubWidth (8-69)

yL = palette\_transpose\_flag ? y \* nSubWidth : y \* nSubHeight (8-70)

with the following:

xL = palette\_transpose\_flag ? y \* nSubHeight : x \* nSubWidth (8-69)

yL = palette\_transpose\_flag ? x \* nSubWidth : y \* nSubHeight (8-70)

1. In clause 9.3.3.1, change the following (note that it was said in the JCTVC-AL meeting minutes that this was confirmed present in H.265 (11/19), but it is not in the published ITU-T H.265/HEVC version 7):

cMax = ( PrefixOffset  <<  1 ) > PaletteMaxRunMinus1 ?  
( PalletMaxRun − PrefixOffset ) : ( PrefixOffset − 1 )

to

cMax = ( PrefixOffset  <<  1 ) > PaletteMaxRunMinus1 ?  
( PalleteMaxRunMinus1 − PrefixOffset ) : ( PrefixOffset − 1 )

1. Consider ticket #1491 Duplicate invocation of 9.3.4.3 arithmetic decoding process (invoked both in 9.3.4.1 and also in 9.3.4.2). Confirmed present in H.265 (11/19) – To be confirmed whether action is needed)
2. In Table 9-43, consider ticket #1498 Typos in Table 9-43 (for palette\_run\_suffix PalletMaxRun should be PalletMaxRunMinus1), Confirmed present in H.265 (11/19).
3. In clause A.4.1, Table A.8, add "MaxSliceSegments" after "Max slice segments per picture".
4. In clause A.4.2, Table A.10, some of the numbers, e.g., "xxxx" are not in the form of "x xxx". Make the style of all the numbers consistent.
5. In clause D.3.1, replace the following:

The list SingleLayerSeiList is set to consist of the payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 134 to 152, inclusive, 154 to 159, and 200 to 202, inclusive.

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

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

with the following:

The list SingleLayerSeiList is set to consist of the payloadType values 2, 3, 6, 9, 15, 16, 17, 19, 22, 23, 45, 47, 56, 128, 129, 131, 132, 134 to 152, inclusive, 154 to 159, inclusive, and 200 to 202, inclusive.

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

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

1. In clause D.3.32, NOTE 2, some of the numbers, e.g., "1000" are not in the form of "1 000". Make the style of all the numbers consistent.
2. In clause D.3.39, check whether the values "0.0001" and "0.00002" should be in the forms of "0.000 1" and "0.000 02", respectively.
3. In the ITU-T version of HEVC and the w18881 MPEG document for FDAM1, in the semantics of ar\_object\_label\_update\_flag, “will not” is repeated.
4. In D.3.41.6.2 (Projection for one sample location), it says that there are inputs to the process that are “pictureWidth and pictureHeight, which are the width and height, respectively, of a monoscopic projected luma picture, in relative projected picture sample units” and “the centre point of a sample location (hPos, vPos) along the horizontal and vertical axes, respectively, in relative projected picture sample units, where hPos and vPos may have non-integer real values”. It does not seem clear what this means.
5. The pictureWidth and pictureHeight variables are also modified internally in D.3.41.6.2, which seems potentially unwise.
6. D.3.41.6.4 (Conversion of sample locations for rectangular region-wise packing) has inputs that are “offset values for the sampling position (offsetX, offsetY)”. D.3.41.6.5 says “offsetX is set equal to 0.5 and offsetY is set equal to 0.5.” Are those the only values that are ever used? If so, why define a variable that does not vary in value?
7. The variable names offsetX and offset that are used in D.3.41.6.4 are also used in 8.5.3.2, but for a different purpose. It may be unwise to use the same variable names in Annex D.
8. The first sentence of D.3.41.6.5 (Mapping of luma sample locations within a cropped decoded picture to sphere coordinates relative to the global coordinate axes) says “This clause specifies the semantics of luma sample locations within a cropped decoded picture to sphere coordinates relative to the global coordinate axes.” Here, “semantics” should probably be “mapping”.
9. In clause D.3.41.7.2, Equation D-57, change the last row from "θ = Asin( z ÷ Sqrt( x2 + y2 + z2 ) ) ÷ π" to "θ = Asin( z ÷ Sqrt( x2 + y2 + z2 ) ) \* 180 ÷ π".
10. The indent distance for the titles of clauses D.3.41.7.3 to D.3.41.7.6 should be aligned to be with those for clauses D.3.41.7.1, D.3.41.7.2, and D.3.41.7.7.
11. D.3.41.7.7 is only for the fisheye video information SEI message, which is handled somewhat separately and differently than other projection mappings; is this appropriate and clear enough?
12. In clause E.3.1, check whether the values "0.3127" etc. should be in the forms of "0.312 7" etc.
13. In clause E.3.1, replace "matrix\_coeffs shall not be equal to 0 unless one or more of the following conditions are true:" with "matrix\_coeffs shall not be equal to 0 unless both of the following conditions are true:".
14. In clause E.3.1, Equations E60-E67, some numbers are in the forms of "0.986566" and "4096" etc. instead of in the forms of "0.986 566" and "4 096" etc.
15. In clause E.3.1, Table E.5, some numbers are in the forms of "0.0593" etc. instead of in the forms of "0.059 3"etc.
16. In clause E.3.1, NOTE 16, change "1920x540" to "1 920 x 540".
17. In clause E.3.2, replace "The need for presence of buffering periods for NAL HRD operation to be present in the bitstream in buffering period SEI messages is determined by the application" with "The need for the presence of buffering period parameters for NAL HRD operation in the bitstream in buffering period SEI messages is determined by the application".
18. In clause E.3.2, replace "The need for presence of buffering periods for VCL HRD operation to be present in the bitstream in buffering period SEI messages is determined by the application" with "The need for the presence of buffering period parameters for VCL HRD operation in the bitstream in buffering period SEI messages is determined by the application".
19. In clause E.3.2, replace "The need for presence of CPB and DPB output delays to be present in the bitstream in picture timing SEI messages is determined by the application" with "The need for the presence of CPB and DPB output delays in the bitstream in picture timing SEI messages is determined by the application".
20. In clause F.7.4.3.1.1, replace "1023" with "1 023".
21. In clauses F.13.2.3 and F.13.4, replace "90000" with "90 000".
22. In clause G.14.3.1, replace the following

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

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

with the following:

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

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

1. In clause I.7.4.3.1.7, replace "65535" with "65 535".
2. In clause I.7.4.3.2.3, replace "is not used the decoding process" with "is not used in the decoding process", and replace "is not present coding units" with "is not present in coding units".
3. In clause I.14.3.1, replace the following:

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

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

with the following:

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

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

# Reported errata items for AVC

See also section 1.

## Publication status background

Rec. ITU-T H.264

* (06/19, Edition 13) Approved 2019-06-13, published 2019-09-06

ISO/IEC 14496-10

* ISO/IEC 14496-10:2014 (Edition 8), published 2014-09
* ISO/IEC 14496-10:2014/Amd 1:2015 (Multi-resolution frame compatible stereoscopic video with depth maps, additional supplemental enhancement information and video usability information), published 2015-11
* ISO/IEC 14496-10:2014/FDAMD 2 (Additional Levels and Supplemental Enhancement Information); stage 50.98, deleted in preparation for Edition 9
* ISO/IEC 14496-10:2014/Amd 3:2016 (Additional supplemental enhancement information); published 2016-12, published 2016-12
* ISO/IEC DIS 14496-10:201x (Edition 9); stage 40.99 Full report circulated: DIS approved for registration as FDIS as of 2018-01-31

## Text bug fixes for Annexes I and J

### Status

These bugs were confirmed, and the text bug fixes were agreed by the JCT-VC at its 37th meeting in Geneva in Oct. 2019. See Section 1 of [JCTVC-AK0022](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10962).

### Bug fixes

*In I.13.2.3.1, change the semantics of da\_mantissa\_len\_minus1 to the following:*

**da\_mantissa\_len\_minus1** + 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.

*In J.7.3.2.13.1, change the depth ranges syntax to the following:*

|  |  |  |
| --- | --- | --- |
| depth\_ranges( numViews, predDirection, index ) { | C | Descriptor |
| **z\_near\_flag** | 11 | u(1) |
| **z\_far\_flag** | 11 | u(1) |
| if( z\_near\_flag ) |  |  |
| 3dv\_acquisition\_element( numViews, predDirection, 7, index, ZNearSign,  ZNearExp, ZNearMantissa, ZNearManLen ) |  |  |
| if( z\_far\_flag ) |  |  |
| 3dv\_acquisition\_element( numViews, predDirection, 7, index, ZFarSign,  ZFarExp, ZFarMantissa, ZFarManLen ) |  |  |
| } |  |  |

*In J.7.3.2.13.2, change the 3DV acquisition element syntax as follows:*

|  |  |  |
| --- | --- | --- |
| 3dv\_acquisition\_element( numViews, predDirection, expLen, index, outSign, outExp, outMantissa, outManLen ) { | **C** | Descriptor |
| if( numViews > 1 ) |  |  |
| **element\_equal\_flag** | 11 | u(1) |
| if( element\_equal\_flag  = =  0 ) |  |  |
| numValues = numViews |  |  |
| else |  |  |
| numValues = 1 |  |  |
| for( i = 0; i < numValues; i++ ) { |  |  |
| if( predDirection  = =  2  &&  i  = =  0 ) { |  |  |
| **mantissa\_len\_minus1** | 11 | u(5) |
| outManLen[ index, i ] = manLen = mantissa\_len\_minus1 + 1 |  |  |
| } |  |  |
| if( predDirection  = =  2 ) { |  |  |
| **sign0** | 11 | u(1) |
| outSign[ index, i ] = sign0 |  |  |
| **exponent0** | 11 | u(v) |
| outExp[ index, i ] = exponent0 |  |  |
| **mantissa0** | 11 | u(v) |
| outMantissa[ index, i ] = mantissa0 |  |  |
| } else { |  |  |
| **skip\_flag** | 11 | u(1) |
| if( skip\_flag = = 0 ) { |  |  |
| **sign1** | 11 | u(1) |
| outSign[ index, i ] = sign1 |  |  |
| **exponent\_skip\_flag** | 11 | u(1) |
| if( exponent\_skip\_flag = = 0 ) { |  |  |
| **exponent1** | 11 | u(v) |
| outExp[ index, i ] = exponent1 |  |  |
| } else |  |  |
| outExp[ index, i ] = outExp[ ref\_dps\_id0, i ] |  |  |
| **mantissa\_diff** | 11 | se(v) |
| if( predDirection = = 0 ) |  |  |
| mantissaPred = (( outMantissa[ ref\_dps\_id0, i ] \* predWeight0 +  outMantissa[ ref\_dps\_id1, i ] \* ( 64-predWeight0 ) + 32 ) >> 6 ) |  |  |
| else |  |  |
| mantissaPred = outMantissa[ ref\_dps\_id0, i ] |  |  |
| outMantissa[ index, i ] = mantissaPred + mantissa\_diff |  |  |
| outManLen[ index, i ] = outManLen[ ref\_dps\_id0, i ] |  |  |
| } else { |  |  |
| outSign[ index, i ] = outSign[ ref\_dps\_id0, i ] |  |  |
| outExp[ index, i ] = outExp[ ref\_dps\_id0, i ] |  |  |
| outMantissa[ index, i ] = outMantissa[ ref\_dps\_id0, i ] |  |  |
| outManLen[ index, i ] = outManLen[ ref\_dps\_id0, i ] |  |  |
| } |  |  |
| } |  |  |
| } |  |  |
| if( element\_equal\_flag = = 1 ) { |  |  |
| for( i = 1; i < numViews; i++ ) { |  |  |
| outSign[ index, i ] = outSign[ index, 0 ] |  |  |
| outExp[ index, i ] = outExp[ index, 0 ] |  |  |
| outMantissa[ index, i ] = outMantissa[ index, 0 ] |  |  |
| outManLen[ index, i ] = outManLen[ index, 0 ] |  |  |
| } |  |  |
| } |  |  |
| } |  |  |

*In J.7.4.2.13.2, change the 3DV acquisition element semantics as follows:*

The syntax structure specifies the value of an element in the depth ranges syntax structure. The element may contain one or more loop entries i of the order specified by view\_id\_3dv syntax elements.

The contents of the syntax structure are controlled through input variables predDirection, expLen, and index, the semantics of which are as follows.

– predDirection equal to 2 specifies that the first loop entry of the element is not predicted and coded in the sign, exponent, and mantissa syntax elements. predDirection equal to 0 or 1 specifies that the first loop entry of the element is predicted and a difference relative to a prediction value is coded in the difference syntax element.

– expLen specifies the number of bits in the exponent syntax element.

– index greater than 0 specifices the depth\_parameter\_set\_id of the depth parameter set wherein the parameters are present, and index equal to 0 specifies that the parameters are present in a sequence parameter set.

The syntax structure uses outSign, outExp, outMantissa, and outManLen variables for both input and output, where each variable is indexed by [ index, viewIdc ], index being an identifier (equal to either 0 when decoding depth ranges in sequence parameter set or depth\_parameter\_set\_id value when decoding depth range parameter set) to a depth parameter set and viewIdc being a view indicator (in the order of views for 3DV acquisition parameters).

**element\_equal\_flag** equal to 0 specifies that the sign, exponent, and mantissa may or may not be identical to respective values for any two loop entries i and j. element\_equal\_flag equal to 1 specifies that the sign, exponent, and mantissa are identical to respective values for any two loop entries i and j. When not present, element\_equal\_flag is inferred to be equal to 0.

**mantissa\_len\_minus1** plus 1 specifies the number of bits in the mantissa syntax element. The value of mantissa\_len\_minus1 shall be in the range of 0 to 31, inclusive.

**sign0** equal to 0 indicates that the sign of the value provided in the loop entry is positive. sign0 equal to 1 indicates that the sign is negative.

**exponent0** specifies the exponent of the value provided by the loop entry. The syntax element exponent0 is represented by expLen bits. The value of exponent0 shall be in the range of 0 to 2expLen – 2, inclusive. The value 2expLen – 1 is reserved for future use by ITU‑T | ISO/IEC. Decoders shall treat the value 2expLen – 1 as indicating an unspecified value.

**mantissa0** specifies the mantissa of the value provided by the loop entry. The syntax element mantissa0 is represented by manLen bits.

**skip\_flag** equal to 0 specifies that syntax elements sign1, exponent\_skip\_flag and mantissa\_diff are present for the loop entry. skip\_flag equal to 1 specifies that elements sign1, exponent\_skip\_flag and mantissa\_diff are not present for the loop entry.

**sign1** equal to 0 indicates that the sign of the value provided in the loop entry is positive. sign1 equal to 1 indicates that the sign is negative.

**exponent1**, if present, specifies the exponent of the value provided by the loop entry. The syntax element exponent1 is represented by expLen bits. The value of exponent1 shall be in the range of 0 to 2expLen – 2, inclusive. The value 2expLen – 1 is reserved for future use by ITU‑T | ISO/IEC. Decoders shall treat the value 2expLen – 1 as indicating an unspecified value.

**mantissa\_diff** specifies the difference of the mantissa of the value provided by the loop entry relative to its prediction value.

## On semantics of nal\_hrd\_parameters\_present\_flag and vcl\_hrd\_parameters\_present\_flag

### Status

These bugs were confirmed, and the text bug fixes were agreed by the JCT-VC at its 37th meeting in Geneva in Oct. 2019. See Section 2 of [JCTVC-AK0022](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10962).

### Bug fixes

*Change the semantics of nal\_hrd\_parameters\_present\_flag and vcl\_hrd\_parameters\_present\_flag as follows (additions are yellow-highlighted):*

**nal\_hrd\_parameters\_present\_flag** equal to 1 specifies that NAL HRD parameters (pertaining to the Type II bitstream conformance point) are present. nal\_hrd\_parameters\_present\_flag equal to 0 specifies that NAL HRD parameters are not present.

NOTE 12 – When nal\_hrd\_parameters\_present\_flag is equal to 0, the conformance of the bitstream cannot be verified without provision of the NAL HRD parameters and all buffering period SEI messages, and, when vcl\_hrd\_parameters\_present\_flag is also equal to 0, all picture timing SEI messages, by some means not specified in this Recommendation | International Standard.

...

**vcl\_hrd\_parameters\_present\_flag** equal to 1 specifies that VCL HRD parameters (pertaining to the Type I ~~all~~ bitstream conformance point) are present. vcl\_hrd\_parameters\_present\_flag equal to 0 specifies that VCL HRD parameters are not present.

NOTE 13 – When vcl\_hrd\_parameters\_present\_flag is equal to 0, the conformance of the bitstream cannot be verified without provision of the VCL HRD parameters and all buffering period SEI messages, and, when nal\_hrd\_parameters\_present\_flag is also equal to 0, all picture timing SEI messages, by some means not specified in this Recommendation | International Standard.

...

## On semantics of rbsp\_byte[ i ]

### Status

These bugs were confirmed, and the text bug fixes were agreed by the JCT-VC at its 37th meeting in Geneva in Oct. 2019. See [JCTVC-AK0023](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10963).

### Bug fixes

*Change the semantics of rbsp\_byte[ i ] to the following:*

**rbsp\_byte**[ i ] is the i-th byte of an RBSP. An RBSP is specified as an ordered sequence of bytes as follows:

The RBSP contains a string of data bits (SODB) as follows:

– If the SODB is empty (i.e., zero bits in length), the RBSP is also empty.

– Otherwise, the RBSP contains the SODB as follows:

1) The first byte of the RBSP contains the first (most significant, left-most) eight bits of the SODB; the next byte of the RBSP contains the next eight bits of the SODB, etc., until fewer than eight bits of the SODB remain.

2) The rbsp\_trailing\_bits( ) syntax structure is present after the SODB as follows:

i) The first (most significant, left-most) bits of the final RBSP byte contain the remaining bits of the SODB (if any).

ii) The next bit consists of a single bit equal to 1 (i.e., rbsp\_stop\_one\_bit).

iii) When the rbsp\_stop\_one\_bit is not the last bit of a byte-aligned byte, one or more zero-valued bits (i.e., instances of rbsp\_alignment\_zero\_bit) are present to result in byte alignment.

3) One or more cabac\_zero\_word 16-bit syntax elements equal to 0x0000 may be present in some RBSPs after the rbsp\_trailing\_bits( ) at the end of the RBSP.

## On semantics of num\_sample\_shift\_plus512[ i ]

At the 38th JCT-VC meeting in 2020-01, the following issue was reported by P. Wu.

The semantics of num\_sample\_shift\_plus512[ i ] says:

**num\_sample\_shift\_plus512[**i ]indicates the recommended additional horizontal shift for a stereo-pair corresponding to the i-th reference baseline and the i-th reference display. If ( num\_sample\_shift\_plus512[ i ] − 512 ) is less than 0, it is recommended that the left view of the stereo-pair corresponding to the i-th reference baseline and the i-th reference display is shifted in the left direction by ( 512 − num\_sample\_shift\_plus512[ i ] ) samples with respect to the right view of the stereo-pair; if num\_sample\_shift\_plus512[ i ] is equal to 512, it is recommended that shifting is not applied; **if ( num\_sample\_shift\_plus512[ i ] − 512 ) is greater than 0**, it is recommended that the left view in the stereo-pair corresponding to the i-th reference baseline and the i-th reference display should be shifted in the right direction by **( 512 − num\_sample\_shift\_plus512[ i ] )** samples with respect to the right view of the stereo-pair. The value of num\_sample\_shift\_plus512[ i ] shall be in the range of 0 to 1023, inclusive.

If ( num\_sample\_shift\_plus512[ i ] < 512 ) , then ( 512 − num\_sample\_shift\_plus512[ i ] ) is a negative value. The correct value is reported to be ( num\_sample\_shift\_plus512[ i ]−512 ).

In the HEVC standard, this problem had previously existed but the text has since been corrected

# Reported errata items for Video CICP

See section 1.

## Publication status background

Rec. ITU-T H.273

* (02/16, Edition 1) Approved 2016-12-22, published 2017-04-27

ISO/IEC 23091-2 (previously part of ISO/IEC 23001-8)

* ISO/IEC 23091-2:2019, published 2019-07

## Errata items

1. There is an error in the range of values for the sample aspect ratio width and height.
2. The CICP revision should account for the various errata issues previously noted in AL1004. It was confirmed that the BT.2100 reference needs updating (to -2).
3. The equations for ICtCp with HLG have not been updated and need be corrected.

# Reported errata items for Codepoint Usage TR

## Publication status background

Sup. ITU-T H.Sup19

* (03/19, Edition 1) Approved 2019-03, published 2019-04-30
* (10/19, Edition 2) Approved 2019-10, published 2019-11-14

ISO/IEC 23091-4

* ISO/IEC 23091-4:2019, published 2019-08
* ISO/IEC 23091-4:202x, stage 60.00 Under publication since 2020-04-05

## Miscellaneous issues

1. In a liaison letter sent to ITU-T SG16, SMPTE identified an anomaly in Table 7 of the 1st edition, as follows:

|  |  |  |
| --- | --- | --- |
| Mastering Display Minimum Luminance | Registration identifier | was  060e2b34.0101010e.04200401.01030000  should be  urn:smpte:ul:060e2b34.0101010e.04200401.01040000  see MasteringDisplayMaximumLuminance at  https://registry.smpte-ra.org/view/published/elements\_by\_group\_view.html |

1. SMPTE also recommended using urn notation for writing labels within a text “urn:smpte:ul:”, as they said this significantly increases the ability to machine parse and process documents. The correction to the supplied document was provided in the urn notation above.
2. The registers can be checked online at this URL: <https://registry.smpte-ra.org/apps/pages/published/>. This could be mentioned in the text.
3. For 3.14 of the ISO/IEC pre-publication text, an editor suggested that the definition of “narrow range” does not describe narrow range sufficiently, and suggested defining it as “range in a fixed-point (integer) representation where active video does not span the full range of values that could be expressed with that bit depth but does use the remaining range for under/overshoot (processing artifacts) and sync”
4. In clause 4 of the ISO/IEC pre-publication text, an editor suggested that UHD now informally includes 1080P with HDR and WCG, and proposed the following “UHD: informally used with several resolutions (3840 × 2160 for television or 4096 × 2160 for film with or without HDR/WCG); (7680 × 4320 or 8192 × 4320 with or without HDR/WCG); (1080P WITH HDR/WCG)”.
5. In the last paragraph of clause 6 of the ISO/IEC pre-publication text, an editor suggested to add MaxFall/MaxCLL to make a sentence more complete, as in “The approved colour volume, which may be smaller than the container volume, is often indicated with SMPTE ST 2086 including MaxFall/MaxCLL metadata. Over time, it is expected that WCG and/or high dynamic range (HDR) applications will evolve to use more of the available container colour volume.”
6. For 7.1 of the ISO/IEC pre-publication text, an editor suggested to add facts: that MP4 containers (and QuickTime) use color atoms utilizing MPEG CICP values to indicate Color Gamut/Transfer Function/Matrix Coefficients, as in “MP4 (and QuickTime) wrappers utilize MPEG CICP integer values shown in Table 4 to indicate Color Gamut/Transfer Function/Matrix Coefficients. These values are stored in color “atoms”.”
7. For an Annex C of the ISO/IEC pre-publication text, an editor suggested to consider adding a reference to RP 214 for carriage of dynamic HDR metadata, saying that RP 214 is being considered for storage of dynamic HDR metadata like ST.2094 and that dynamic metadata can carry more information about content within a stream which assists in remapping HDR video images to a specific displays capabilities. RP 214 describes a means for packing SMPTE KLV encoded metadata and/or essence into SMPTE 291M ancillary data packets. The editor said there is strong interest in carrying audio/visual data, metadata, and/or essence, within the same digital stream and said that the use of one data stream facilitates delivery of the overall multimedia presentation. Metadata is classified as information about the essence.
8. Check for P3D65x1000n0005 vs P3D65x1000n005
9. Check for P3D65x4000n0005 vs P3D65x4000n005
10. Check for “specs"” (ending in a quote mark)
11. Review the phrase “and the combination is specified in ITU-R and/or SMPTE specifications”
12. 7.2.3 note below table 5 add “(Rec. ITU-R BT.709)” to the sentence: "ATSC specifications list use of the transfer characteristics value of 1 for SDR NCG  (Rec. ITU-R BT.709) video"
13. A small punctuation suggestion for 7.2.4 (adding a comma):

“It is important for tools to process video according to the colour volume it is operating in to make sure the conversion is consistent.”

Change to:

“It is important for tools to process video according to the colour volume it is operating in, to ensure the conversion is consistent.”

1. A slightly awkward sentence in the same note area in 7.2.4:

"ARIB STD B32 lists use of the transfer characteristics value 1 for HD and 14 for UHD for SDR WCG video.”

Change to:

"ARIB STD B32 lists use of the transfer characteristics value 1 for HD (Rec. ITU-R BT.709) and 14 for UHD SDR WCG (Rec. ITU-R BT.2020) video.”

1. To the last sentence in 7.2.4, add as in

“The indicated chroma sample location alignment is only applicable for 4:2:0 chroma sampling. ChromaLocType (the generic label used in this document for the HEVC and AVC bitstream syntax elements: chroma\_sample\_loc\_type\_top\_field and chroma\_sample\_loc\_type\_bottom\_field), listed in Tables 1 and 3 of this document, indicates the 4:2:0 chroma sample position alignment.”

Change to:

“The indicated chroma sample location alignment is only applicable for 4:2:0 chroma sampling. ChromaLocType (the generic label used in this document for the HEVC and AVC bitstream syntax elements: chroma\_sample\_loc\_type\_top\_field and chroma\_sample\_loc\_type\_bottom\_field), listed in Tables 1 and 3 of this document, indicates the 4:2:0 chroma sample position alignment.  The Rec. ITU-R program signal exchange series (BT.601, BT.709, BT.2020, BT.2100) consider 4:2:2 and 4:4:4 chroma samples to be co-sited with luma.”

1. Also insert parenthetic details in Annex A

“The transfer characteristics indicator values of 1, 6, 14, and 15 are functionally the same. Blu-ray BD-ROM 3.1 (“4K”) and the DVB UHD specifications list use of the transfer characteristics value of 14 for SDR/WCG (Rec. ITU-R BT.2020) video. ATSC specifications list use of the transfer characteristics value of 1 for SDR video. ARIB STD B32 lists use of the transfer characteristics value 1 for HD and 14 for UHD for SDR WCG video.”

Change to:

“The transfer characteristics indicator values of 1, 6, 14, and 15 are functionally the same. Blu-ray BD-ROM 3.1 (“4K”) and the DVB UHD specifications list use of the transfer characteristics value of 14 for SDR/WCG (Rec. ITU-R BT.2020) video. ATSC specifications list use of the transfer characteristics value of 1 for SDR NCG (Rec. ITU-R BT.709) and SDR WCG (Rec. ITU-R BT.2020) video. ARIB STD B32 lists use of the transfer characteristics value 1 for HD (Rec. ITU-R BT.709) and 14 for UHD SDR WCG (Rec. ITU-R BT.2020) video.”

1. General: Review parentheticals
2. General: Review/rephrase instances of “and that”.