

**ISO/IEC JTC 1/SC 29/WG 11 N 18711**

**ISO/IEC JTC 1/SC 29/WG 11**

**Coding of moving pictures and audio**

**Convenorship: UNI (Italy)**

|  |  |
| --- | --- |
| **Document type:** | **Approved WG 11 document** |
|  |  |
| **Title:** | **Defect report for HEVC (ISO/IEC 23008-2), AVC (ISO/IEC 14496-10), and Video CICP (ISO/IEC 23091-2)** |
|  |  |
| **Status:** | **Approved** |
|  |  |
| **Date of document:** | **2019-07-12** |
|  |  |
| **Source:** | **WG 11 (via JCT-VC)** |
|  |  |
| **Expected action:** | **Study** |
|  |  |
| **No. of pages:** | **9** |
|  |  |
| **Email of convenor:** | **leonardo@chiariglione.org** |
|  |  |
| **Committee URL:** | **https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg11** |

|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  36th Meeting: Gothenburg, SE, 6–12 July 2019 | Document: JCTVC-AJ1004-v1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Errata report items for HEVC, AVC, and Video CICP** | | |
| *Status:* | Output document approved by JCT-VC | | |
| *Purpose:* | Errata report | | |
| *Author(s) or Contact(s):* | **Gary Sullivan**  **Ye-Kui Wang** | Tel: Email: | [garysull@miscrosoft.com](mailto:garysull@miscrosoft.com)  [ye-kui.wang@futurewei.com](mailto:ye-kui.wang@futurewei.com) |
| *Source:* | Editors | | |

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

# Abstract

This document contains a list of reported errata items for HEVC, AVC, and Video CICP, 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:

* Additional aspects of 3D-AVC (see section 3.1)
* The editorial phrasing relating to encapsulation of an SODB in an RBSP (esp. in AVC)
* A sentence in subclause 0.5 in Rec. ITU-T H.265 2019-06 as pre-published says it is the fourth version, but later describes six versions and says the current version is the sixth version; clearly it should say it is the sixth version. (The sentence saying it is the fourth version should just be removed since it is redundant with what is stated later and could cause a similar error for future editions, as it did in the published fifth version as well.)
* Alphabetical ordering of the definitions of "associated IRAP picture" and "associated non-VCL NAL unit" (incorrect in Rec. ITU-T H.265 2019-06 as pre-published).
* Indentation at the end of subclause 8.6.8.2 (Adaptive colour transformation process) in HEVC (incorrect in Rec. ITU-T H.265 2019-06 as pre-published).
* 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 as pre-published).
* Logical indentation nesting in the ISO/IEC version of the Video CICP colour interpretation was inadvertently removed in the publication process.
* 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

In an initial review by the JCT-VC, this report seemed to be correct. It probably indicates a need for correction in HEVC, AVC, and CICP (and JPEG XR). The suggested correction requires further study to determine the appropriate way to address the issue. For background, see [JCTVC-AJ0023](http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=10941).

### Description of the issue

An errata report was submitted regarding the text relating to the transfer characteristics and matrix coefficients in regard to the sYCC colour representation specified in IEC 61966-2-1. It asserts that the semantics of transfer characteristics (Table E.4 of HEVC), and the semantics of matrix coefficients (Table E.5 of HEVC) need correction.

### Description of the proposed fix

The first change proposed in the document contribution is to change the interpretation of the transfer\_characteristics syntax element for the value 13 from:

V = *α* \* Lc( 1 ÷ 2.4 ) − ( *α* − 1 ) for 1 >= Lc >= *β*

V = 12.92 \* Lc for *β* > Lc >= 0

(With the informative remark of "IEC 61966-2-1 sRGB or sYCC")

to:

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 Lc < −β

and to also place an asterisk next to the mention of sRGB and add an informative NOTE saying that Lc is clipped between 0 and 1 for sRGB, as follows:

NOTE – For transfer\_characteristics equal to 13, Lc values between 0 and 1are allowed for sRGB.

However, this proposed fix only discusses the allowed sRGB range informatively, and does not express a very clear way to distinguish between the cases where the clipping is considered to be applied and when it is not. It also redefines the interpretation of the existing specification. An alternative possibility could be 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). A third possibility could be to assign a new value of the transfer\_characteristics syntax element for sYCC, but this would result in the failure of existing implementations to recognize and interpret the indication.

The second change proposed in the document contribution is to change the informative remark relating to IEC 61966-2-1 sYCC to indicate that it should be indicated with the matrix\_coeffs syntax element equal to 5 (as for Rec. ITU R BT.601) rather than 1 (as for Rec. ITU-R BT.709).

# 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, pre-published 2019-07-17

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
* 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 2019-02-19; Pending FDIS ballot based on WG 11 N 18277 as of 2019-07-07
* ISO/IEC 23008-2:201x (Edition 4)/DAmd 1:201x (Additional supplemental enhancement information); DAM ballot started 2019-07-10, closing 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).

### 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 NOT yet been confirmed by the JCT-VC. For background, see [JCTVC-AJ0026](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10944).

### 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 as follows:

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 )

# 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,

## Mismatch of text and software for SPS MVCD extension for 3D-AVC

### Status

Preliminary study of this issue by the JCT-VC has indicated that the problem is likely to be the software rather than the text. Further study of issues relating to 3D-AVC is ongoing. For background, see [JCTVC-AI0002](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10932) and MPEG input document [M47286](http://phenix.it-sudparis.eu/mpeg/doc_end_user/current_document.php?id=67137&id_meeting=178).

### Description of the issue

There was a report of a potential mismatch between text and software. It was reported, through SC 29 Committee Manager Mayumi Koike, on February 18, 2019, that someone said there was a discrepancy between the published H.264 standard and the 3DAVC reference software.

Considering section I.7.3.2.1.5 of H.264 (04/17), below is the syntax of seq\_parameter\_set\_mvcd\_extension():

|  |  |  |
| --- | --- | --- |
| seq\_parameter\_set\_mvcd\_extension( ) { | **C** | **Descriptor** |
| **num\_views\_minus1** | 0 | ue(v) |
| for( i = 0, NumDepthViews = 0; i <= num\_views\_minus1; i++ ) { |  |  |
| **view\_id**[ i ] | 0 | ue(v) |
| **depth\_view\_present\_flag**[ i ] | 0 | u(1) |
| DepthViewId[ NumDepthViews ] = view\_id[ i ] |  |  |
| NumDepthViews += depth\_view\_present\_flag[ i ] |  |  |
| **texture\_view\_present\_flag**[ i ] | 0 | u(1) |
| } |  |  |
| …continued | | |

The reference software in ISO IEC 14496-5 2001 Amd 35 reportedly does not correspond to this specification (file ISO IEC 14496-5 2001 Amd 35\3DV-ATM\_v14.0\lencod\src\parset.c):

if(sps->profile\_idc==ThreeDV\_HIGH )

{

if (p\_Vid->is\_depth==0)

{

len+=u\_1 ("SPS: depth\_view\_present\_flag", p\_Vid->p\_DualInp->ViewPresentFlag[i], bitstream);

len+=u\_1 ("SPS: texture\_view\_present\_flag", p\_Inp->ViewPresentFlag[i], bitstream);

}

else

{

len+=u\_1 ("SPS: depth\_view\_present\_flag", p\_Inp->ViewPresentFlag[i], bitstream);

len+=u\_1 ("SPS: texture\_view\_present\_flag", p\_Vid->p\_DualInp->ViewPresentFlag[i], bitstream);

}

}

}

It was reported that in the specification, the syntax elements depth\_view\_present\_flag and texture\_view\_present\_flag are present unconditionally, while the reference software writes these flags only if profile\_idc is equal 138 (Multiview Depth High profile).

# 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