|  |
| --- |
| **INTERNATIONAL ORGANIZATION FOR STANDARDIZATION ORGANISATION INTERNATIONALE DE NORMALISATION ISO/IEC JTC 1/SC 29/WG 5 MPEG JOINT VIDEO CODING TEAM(S) WITH ITU-T SG 16** |
| **ISO/IEC JTC 1 / SC 29 / WG 5 N 242** |
| **Hannover, DE – 13–20 October 2023** |
| |  |  | | --- | --- | | **Source:** | **Convenor (Jens-Rainer Ohm)** | | **Title:** | **Preliminary working draft 6 of SEI processing order and processing order nesting SEI messages in VVC** | | **Type:** | **General** | | **Subtype:** | **Other** | | **Status:** | **Approved** | | **Date:** | **2024-01-19** | | **Expected Action:** | **Info** | | **Action due date:** | **N/A** | | **Pages:** | **50** (not including this cover page) | | **Email of convenor:** | **ohm @ ient . rwth-aachen . de** | | **Committee URL:** | **https://sd.iso.org/documents/ui/#!/browse/iso/iso-iec-jtc-1/iso-iec-jtc-1-sc-29/iso-iec-jtc-1-sc-29-wg-5** | |

|  |  |
| --- | --- |
| **Joint Video Experts Team (JVET)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29**  32nd Meeting, Hannover, DE, 13–20 October 2023 | Document: JVET-AF2032-v4 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Technologies under consideration for future extensions of VSEI (version 2)** | | |
| *Status:* | Output document approved by JVET | | |
| *Purpose:* | Draft text | | |
| *Author(s) or Contact(s):* | Miska M. Hannuksela Jie Chen Sachin Deshpande Hendry Sean McCarthy | Email: | miska.hannuksela@nokia.com jiechen.cj@alibaba-inc.com [sdeshpande@sharplabs.com](mailto:sdeshpande@sharplabs.com) dr.hendry@lge.com sean.mccarthy@dolby.com |
| *Source:* | Editors | | |

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

# Abstract

This document contains the draft text for changes under consideration for future extensions to the versatile supplemental enhancement information messages for coded video bitstreams (VSEI) standard (Rec. ITU-T H.274 | ISO/IEC 23002-7) to modify existing SEI messages or specify additional SEI messages. This document also contains the draft text for changes under consideration for future extensions to the versatile video coding (VVC) standard (Rec. ITU-T H.266 | ISO/IEC 23090-3) to for specifying the use of SEI messages specified in VSEI in VVC bitstreams.

This version of the document contains draft text changes for adding the support of grouping of NNPFs through modifying the neural-network post-filter SEI messages and adding neural-network post-filter group SEI messages. This version of the document also contains an alternative set of draft text changes for adding the support of grouping of both NN based and non-NN-based post-processing filters (PFs) through modifying the semantics of the neural-network post-filter SEI messages and using the po\_id of the SEI processing order (SPO) SEI messages.

This version of the document also contains draft text for addition of new SEI messages, including the neural-network post-filter gain SEI message, the encoder optimization information SEI message, the source picture timing information SEI message, and the object mask information SEI message. Additionally, this version of the document further contains draft text to specify Exif, JFIF, and XMP and multiplane image information SEI messages, signalling of picture modality type in VUI parameters, and large SEI message syntax structure.

For many of the text changes in this version of the document, the base text is JVET-AE2032. However, for some of the text changes, the base text is VSEI 2nd edition, and for some other text changes, the base text is VVC 2nd edition.

**Changes yet to be integrated:**

Summary: changes noted below corresponding to aspects of JVET-AF0055 and JVET-AF0068

1. Items on the source picture timing information (SPTI) SEI message, from JVET-AF0069 [Dolby]
2. Replace syntax element spti\_source\_timing\_equals\_output\_timing\_flag with spti\_source\_timing\_info\_present\_flag and add corresponding semantics. [*Ed: This aspect 8.i appears to conflict with 8.a. Thus, no action has been taken on aspect 8.i*]

**Changes that have been integrated:**

Summary: changes noted below corresponding to JVET-AF0050, JVET-AF0051, JVET-AF0052, JVET-AF0055 (except item 7.c), JVET-AF0061, JVET-AF0069 (except item 8.i), JVET-AF0088, JVET-AF0091, JVET-AF0093, JVET-AF0094, JVET-AF0097, JVET-AF0107, JVET-AF0141, JVET-AF0147, JVET-AF0167

1. Changes in JVET-AF0061 other than the addition of po\_id [Bytedance]
   1. The post-processing filter (PPF) grouping concept based on po\_id in the SPO SEI message, taking into account backward compatibitiy with v3 of VVC/VSEI for the a special PPF cascading case
   2. Updates to the filtering process to apply to both NNPFs and non-NN-based PPFs and to enable the use of multiple activated PPFs from one PPF group in the cascading manner
   3. Updates to the NNPFC semantics in VSEI
   4. Updates to the NNPFC/NNPFA interface text in VVC
2. 2 items from JVET-AF0050 [Nokia]
   1. Addition of nnpfga\_no\_prev\_clvs\_flag and nnpfga\_no\_foll\_clvs\_flag
   2. Removal of the possibility to reference NNPF group identifiers from the NNPFA SEI message
3. The approach of separate SEI (applicable for any PF) from JVET-AF0051, for signalling of the gain provided by post-processing filter(s). Remove the URI, define PSNR in a bit-depth neutral. [Nokia]
4. Items 1, 2 and 3 in JVET-AF0091 [LGE]
   1. Adding the definitions of direct and indirect members of an NNPFGC
   2. Clarifying that the value of nnpfgc\_purpose in NNPFGC, when present, includes all purposes of NNPFC or NNPFGC included in the NNPFGC
   3. When nnpfgc\_grouping\_type is equal to 1 (i.e., alternative grouping) or 3 (i.e., parallel grouping), purpose information is signalled for each member of the NNPFGC.
5. On intermediary output picture(s) from activation of an NNPFGC, from JVET-AF0093 [LGE]
6. Some text clarification of the intent on activation of an NNPFGC that contains another NNPFGC could be useful, as a result from the discussion of JVET-AF0094. [LGE]
7. Some of the items on the source picture timing information (SPTI) SEI message, from JVET-AF0055 [Sharp]
8. Item 1: Add a NOTE.
9. Item 2: Agreed that conditioning the syntax element “spti\_max\_sublayer\_minus1” on persistence is appropriate.
10. Item 3: Editorial action to clarify that the interval at highest layer is not necessarily 1.
11. Item 5: Addition of text for “Use of source picture timing information” for the relevant standards.
12. Item 6 (Syntax rearrangement is proposed such that the syntax element spti\_source\_picture\_timing\_type is signalled only when spti\_source\_timing\_equals\_output\_timing\_flag is equal to 0.): Agreed (in the context of discussing JVET-AF0069) to adopt item 6, however using the **spti\_source\_type\_present\_flag** for possible gating of the syntax element.
13. Items on the source picture timing information (SPTI) SEI message, from JVET-AF0069 [Dolby]
    1. Item 1: It was agreed that the name of the previous spti\_source\_timing\_equals\_output\_timing\_flag should not be modified into spti\_source\_timing\_info\_present\_flag. If it is zero, the spti\_source\_type shall be sent (with its gating flag).
    2. Simplify syntax element name spti\_source\_picture\_timing\_type to spti\_source\_type.
    3. Change the precision of spti\_source\_type from u(8) to u(16).
    4. Simplify the syntax element name spti\_num\_units\_in\_elemental\_source\_picture\_interval to spti\_num\_units\_in\_elemental\_interval.
    5. Simplify the syntax element name spti\_sublayer\_source\_picture\_interval\_scale\_factor[ i ] to spti\_sublayer\_interval\_scale\_factor[ i ].
    6. Change the precision of spti\_num\_units\_in\_elemental\_interval from u(32) to u(18)
    7. Add text to clarify the descriptions of the various source types, e.g., “slow motion”, “high-speed imaging”, etc.
    8. Add semantic constraints to prevent mutually exclusive timing relationships between source pictures and corresponding decoded output pictures. Specifically, prevent the combination of “high-speed imaging” and “time-lapse imaging”.
    9. *[Ed: no action needed on item 8.i]*.
    10. Move specification of the variable temporalReversalFlag to the semantics following spti\_sublayer\_interval\_scale\_factor[ i ].
    11. Integrate the variable temporalReversalFlag in the equation for SourcePictureInterval[ i ] and remove the equation for SourcePictureTime[ i ] (i.e., the absolute source picture time).
14. Add the following text, resulted from JVET-AF0097 item 1 [LGE]:

The information provided by the SPTI SEI message pertains only for picture(s) starting from the picture in the current layer in the access unit that contains the SPTI SEI message and all subsequent pictures of the current layer in output order based on its persistence.

1. Some of the items on the encoder optimization information (EOI) SEI message, from JVET-AF0052 [Nokia]
   1. Item 1: Editorial updates to the phrasing of the cancellation and persistence.
   2. Item 2: A syntax element eoi\_object\_based\_idc to indicate the type of object-based optimization, including blurring, quantization adjustment, and overwriting sample values of areas outside the detected objects.
   3. Item 4: Replacement of the optimization\_ prefix in the syntax element names with eoi\_ to obtain shorter syntax element names.
   4. Item 5: Sensibility constraints that when the persistence is for the current picture only, the temporal optimizations (eoi\_temporal\_subsampling\_flag and eoi\_temporal\_quality\_flag) are required to be off.
   5. Item 6: For temporal subsampling, addition of eoi\_num\_int\_pics, which is indicative of the count of pictures that the encoding systems excluded between each pair of coded pictures in output order.
   6. Item 7: For temporal quality optimization, a clarification of the semantics and addition of a related NOTE.
2. Some of the items on the EOI SEI message, from JVET-AF0107 [LGE]
   1. On re-formulation spatial/temporal resampling from subsampling/downsampling (which would also apply to upsampling).
   2. Simplification of optimization\_type table (editorial) in v2 of JVET-0107 in section 2.1.
   3. It is further reported that in the combination with the adoption from JVET-AF0052 and JVET-AF0107, a discrepancy was found that is resolved in the green highlighted parts of v2 of JVET-0107. It was also agreed to include these changes in the updates of the EOI SEI in the TuC.
3. Items on the object mask information (OMI) SEI message, from JVET-AF0088 [Alibaba]
   1. Aspect 1: The text related to bounding box parameters was fixed and refined.
   2. Aspect 2: The binarization of bounding box parameters was changed from ue(v) to u(16)
   3. Aspect 3: A gating flag for bounding box parameters was added to give signaling flexibility to the encoder
   4. Aspect 4: The parsing dependency among different OMI SEI messages was removed by always signaling omi\_mask\_cancel[ i ][ j ][ k ]…
4. SEI messages for image metadata formats Exif, JFIF, and XMP, from JVET-AF0141, with an appropriate editor’s note on how these formats could be referenced in an ITU-T/ISO standard. [Tencent]
5. Signal picture modality type in VUI parameters, from JVET-AF0147. [Panasonic]
6. The lsei\_message( ) syntax structure for carriage of information about an SEI payload and the SEI payload itself, from JVET-AF0148 [Tencent]
7. The multiplane image information (MPII) SEI message from JVET-AF0167 [Dolby]

**Changes to the specification text:**

*Changes to VVC subclauses 7.3 and 7.4 [Ed. (GJS): Where would this fit into the VVC syntax? Some other syntax structure would need to include it or there would need to be a new NAL unit type and RBSP defined for it.]*

Modify subclauses 7.3 and 7.4 per JVET-AF0148 as follows

7.3.x Large supplemental enhancement information message syntax

|  |  |
| --- | --- |
| lsei\_message( ) { | **Descriptor** |
| **lsei\_position** | u(2) |
| **lsei\_relevance** | u(2) |
| **lsei\_reserved** | u(4) |
| **lsei\_payload\_type\_byte** | u(8) |
| **lsei\_payload\_size\_16bits** | u(16) |
| lsei\_payload( lseiPayloadType, lseiPayloadSize ) |  |
| } |  |

7.4.x Large supplemental enhancement information message semantics

Each large SEI message consists of the variables specifying the type payloadType and size payloadSize of the large SEI message payload. Large SEI message payloads are specified in Annex D. The derived large SEI message payload size payloadSize is specified in bytes and shall be equal to the number of RBSP bytes in the large SEI message payload.

NOTE – The NAL unit byte sequence containing the large SEI message might include one or more emulation prevention bytes (represented by emulation\_prevention\_three\_byte syntax elements). Since the payload size of a large SEI message is specified in RBSP bytes, the quantity of emulation prevention bytes is not included in the size payloadSize of a large SEI payload.

**lsei\_position** indicates if the SEI message corresponds to the PREFIX\_SEI\_NUT and SUFFIX\_SEI\_NUT. **lsei\_position** equal 0 indicates that the SEI message is treated as PREFIX\_SEI\_NUT. **lsei\_position** equal 1 indicates that the SEI message is treated as SUFFIX\_SEI\_NUT. Values 3 and 4 of **lsei\_position** are reserved for future use and shall be ignored.

**lsei\_relevance** indicates the relevance of the SEI message for the target application. **lsei\_relevance** ranges from 0 to 3, with 0 being the least relevant and 3 being the most relevant.

NOTE – The relevance of an SEI message is an arbitrary decision and its use is to be specified by the target application.

**lsei\_reserved** is reserved for future use and shall be ignored.

**lsei\_payload\_type\_byte** is a byte of the payload type of a large SEI message. payloadType = **lsei\_payload\_type\_byte.**

**payload\_size\_16bits** is the payload size in bits of a large SEI message. payloadSize = **payload\_size\_16bits.**

*Changes to VSEI clauses 6 and 7*

Modify clauses 6 and 7 per JVET-AF0147 as follows

6.3 Specification of syntax functions and descriptors

The functions presented in this clause are used in the syntactical description. These functions are expressed in terms of the value of the VUI parameters syntax or an SEI message syntax data pointer that indicates the position of the next bit to be read by the decoding process from the syntax structure.

byte\_aligned( ) is specified as follows:

* If the current position in an SEI message syntax structure or vui\_parameters( ) syntax structure is a byte-aligned position, i.e., the current position is an integer multiple of 8 bits from the position of the first bit in the SEI message syntax structure or vui\_parameters( ) syntax structure, the return value of byte\_aligned( ) is equal to TRUE.
* Otherwise, the return value of byte\_aligned( ) is equal to FALSE.

more\_data\_in\_payload( ) is specified as follows:

– If byte\_aligned( ) is equal to TRUE and the current position in an SEI message syntax structure or vui\_parameters( ) syntax structure is 8 \* payloadSize bits from the beginning of the syntax structure, the return value of more\_data\_in\_payload( ) is equal to FALSE.

– Otherwise, the return value of more\_data\_in\_payload( ) is equal to TRUE.

read\_bits( n ) reads the next n bits from the syntax structure and advances the data pointer by n bit positions. When n is equal to 0, read\_bits( n ) is specified to return a value equal to 0 and to not advance the data pointer.

payload\_extension\_present( ) is specified as follows:

* If the current position in an SEI message syntax structure or vui\_parameters( ) syntax structure is not the position of the last (least significant, right-most) bit that is equal to 1 that is less than 8 \* payloadSize bits from the beginning of the syntax structure, the return value of payload\_extension\_present( ) is equal to TRUE.
* Otherwise, the return value of payload\_extension\_present( ) is equal to FALSE.

The following descriptors specify the parsing process of each syntax element:

– b(8): byte having any pattern of bit string (8 bits). The parsing process for this descriptor is specified by the return value of the function read\_bits( 8 ).

– f(n): fixed-pattern bit string using n bits written (from left to right) with the left bit first. The parsing process for this descriptor is specified by the return value of the function read\_bits( n ).

– i(n): signed integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function read\_bits( n ) interpreted as a two's complement integer representation with most significant bit written first.

– se(v): signed integer 0-th order Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in clause 9 with the order k equal to 0.

– st(v): null-terminated string encoded as universal coded character set (UCS) transmission format-8 (UTF-8) characters as specified in ISO/IEC 10646. The parsing process is specified as follows: st(v) begins at a byte-aligned position in the bitstream and reads and returns a series of bytes from the bitstream, beginning at the current position and continuing up to but not including the next byte-aligned byte that is equal to 0x00, and advances the bitstream pointer by ( stringLength + 1 ) \* 8 bit positions, where stringLength is equal to the number of bytes returned.

NOTE – The st(v) syntax descriptor is only used in this Specification when the current position in the bitstream is a byte-aligned position.

– u(n): unsigned integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function read\_bits( n ) interpreted as a binary representation of an unsigned integer with most significant bit written first.

– ue(v): unsigned integer 0-th order Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in clause 9 with the order k equal to 0.

7.2 VUI parameters syntax

|  |  |
| --- | --- |
| vui\_parameters( payloadSize ) { | **Descriptor** |
| **vui\_progressive\_source\_flag** | u(1) |
| **vui\_interlaced\_source\_flag** | u(1) |
| **vui\_non\_packed\_constraint\_flag** | u(1) |
| **vui\_non\_projected\_constraint\_flag** | u(1) |
| **vui\_aspect\_ratio\_info\_present\_flag** | u(1) |
| if( vui\_aspect\_ratio\_info\_present\_flag ) { |  |
| **vui\_aspect\_ratio\_constant\_flag** | u(1) |
| **vui\_aspect\_ratio\_idc** | u(8) |
| if( vui\_aspect\_ratio\_idc = = 255 ) { |  |
| **vui\_sar\_width** | u(16) |
| **vui\_sar\_height** | u(16) |
| } |  |
| } |  |
| **vui\_overscan\_info\_present\_flag** | u(1) |
| if( vui\_overscan\_info\_present\_flag ) |  |
| **vui\_overscan\_appropriate\_flag** | u(1) |
| **vui\_colour\_description\_present\_flag** | u(1) |
| if( vui\_colour\_description\_present\_flag ) { |  |
| **vui\_colour\_primaries** | u(8) |
| **vui\_transfer\_characteristics** | u(8) |
| **vui\_matrix\_coeffs** | u(8) |
| **vui\_full\_range\_flag** | u(1) |
| } |  |
| **vui\_chroma\_loc\_info\_present\_flag** | u(1) |
| if( vui\_chroma\_loc\_info\_present\_flag ) |  |
| if( vui\_progressive\_source\_flag && !vui\_interlaced\_source\_flag ) |  |
| **vui\_chroma\_sample\_loc\_type\_frame** | ue(v) |
| else { |  |
| **vui\_chroma\_sample\_loc\_type\_top\_field** | ue(v) |
| **vui\_chroma\_sample\_loc\_type\_bottom\_field** | ue(v) |
| } |  |
| if( more\_data\_in\_payload( ) ) { |  |
| if( payload\_extension\_present( ) ) { |  |
| **vui\_modality\_info\_present\_flag** | u(1) |
| if( vui\_modality\_info\_present\_flag ) |  |
| **vui\_modality\_type** | u(8) |
| } |  |
| **vui\_payload\_bit\_equal\_to\_one** /\* equal to 1 \*/ | f(1) |
| while( !byte\_aligned( ) ) |  |
| **vui\_payload\_bit\_equal\_to\_zero** /\* equal to 0 \*/ | f(1) |
| } |  |
| } |  |

7.3 VUI parameters semantics

VUI parameters apply to one or more CLVSs.

NOTE 1 – The interpretation of several syntax elements of the VUI parameters are specified by reference to coding-independent code points specified in Rec. ITU-T H.273 | ISO/IEC 23091-2. Further information about the usage of such code points is found in Supplement ITU-T H-Suppl. 19 | ISO/IEC TR 23091-4

Use of the VUI parameters requires the definition of the following variables:

– A chroma format indicator, denoted herein by ChromaFormatIdc, such that the value 0 indicates that the picture has only a luma component and other values indicate that the picture has three colour components that consist of a luma component and two associated chroma components, such that the width and height of each chroma component are the width and height of the luma component divided by SubWidthC and SubHeightC, respectively, where SubWidthC and SubHeightC are determined from ChromaFormatIdc as specified by Table 2.

– A bit depth for the samples of the luma component, denoted herein by BitDepthY, and when ChromaFormatIdc is not equal to 0, a bit depth for the samples of the two associated chroma components, denoted herein by BitDepthC.

**Table 2 – SubWidthC and SubHeightC values derived from  
ChromaFormatIdc**

|  |  |  |  |
| --- | --- | --- | --- |
| **ChromaFormatIdc** | **Chroma format** | **SubWidthC** | **SubHeightC** |
| 0 | Monochrome | 1 | 1 |
| 1 | 4:2:0 | 2 | 2 |
| 2 | 4:2:2 | 2 | 1 |
| 3 | 4:4:4 | 1 | 1 |

**vui\_progressive\_source\_flag** and **vui\_interlaced\_source\_flag** are interpreted as follows:

– If vui\_progressive\_source\_flag is equal to 1 and vui\_interlaced\_source\_flag is equal to 0, the source scan type of the pictures should be interpreted as progressive only.

– Otherwise, if vui\_progressive\_source\_flag is equal to 0 and vui\_interlaced\_source\_flag is equal to 1, the source scan type of the pictures should be interpreted as interlaced only.

– Otherwise, if vui\_progressive\_source\_flag is equal to 0 and vui\_interlaced\_source\_flag is equal to 0, the source scan type of the pictures should be interpreted as unknown or unspecified or specified by external means not specified in this Specification.

– Otherwise (vui\_progressive\_source\_flag is equal to 1 and vui\_interlaced\_source\_flag is equal to 1), the source scan type of each picture is indicated at the picture level using the syntax element ffi\_source\_scan\_type in a frame-field information SEI message.

**vui\_non\_packed\_constraint\_flag** equal to 1 specifies that there shall not be any frame packing arrangement SEI messages present in the bitstream that apply to the CLVS. vui\_non\_packed\_constraint\_flag equal to 0 does not impose such a constraint.

**vui\_non\_projected\_constraint\_flag** equal to 1 specifies that there shall not be any equirectangular projection SEI messages or generalized cubemap projection SEI messages present in the bitstream that apply to the CLVS. vui\_non\_projected\_constraint\_flag equal to 0 does not impose such a constraint.

**vui\_aspect\_ratio\_info\_present\_flag** equal to 1 specifies that vui\_aspect\_ratio\_idc is present. vui\_aspect\_ratio\_info\_present\_flag equal to 0 specifies that vui\_aspect\_ratio\_idc is not present.

**vui\_aspect\_ratio\_constant\_flag** equal to 1 specifies that the values of vui\_aspect\_ratio\_idc, SarWidth, and SarHeight apply to all pictures in the CLVS and there is no SARI SEI message present in the CLVS. vui\_aspect\_ratio\_constant\_flag equal to 0 specifies that the values of vui\_aspect\_ratio\_idc, SarWidth, and SarHeight might or might not apply to all pictures in the CLVS and that SARI SEI messages could be present in the CLVS indicating a different sample aspect ratio applicable to the pictures associated with SARI SEI messages. When the vui\_aspect\_ratio\_constant\_flag syntax element is not present, the value of vui\_aspect\_ratio\_constant\_flag is inferred to be equal to 0.

**vui\_aspect\_ratio\_idc**, when not equal to 255, indicates the SAR of the luma samples of decoded pictures in the CLVS, unless indicated otherwise by associated SARI SEI messages when vui\_aspect\_ratio\_constant\_flag is equal to 0. Its semantics are as specified for the SampleAspectRatio parameter in Rec. ITU-T H.273 | ISO/IEC 23091-2. When the vui\_aspect\_ratio\_idc syntax element is not present, the value of vui\_aspect\_ratio\_idc is inferred to be equal to 0. Values of vui\_aspect\_ratio\_idc that are specified as reserved for future use in Rec. ITU-T H.273 | ISO/IEC 23091-2 shall not be present in bitstreams conforming to this version of this Specification. Decoders shall interpret values of vui\_aspect\_ratio\_idc that are reserved for future use in Rec. ITU-T H.273 | ISO/IEC 23091-2 as equivalent to the value 0.

**vui\_sar\_width**, when present, indicates the horizontal size of the SAR (in arbitrary units) of the luma samples of decoded pictures in the CLVS, unless indicated otherwise by associated SARI SEI messages when vui\_aspect\_ratio\_constant\_flag is equal to 0.

**vui\_sar\_height**, when present, indicates the vertical size of the SAR (in the same arbitrary units as vui\_sar\_width) of the luma samples of decoded pictures in the CLVS, unless indicated otherwise by associated SARI SEI messages when vui\_aspect\_ratio\_constant\_flag is equal to 0.

When present, vui\_sar\_width and vui\_sar\_height shall be relatively prime or equal to 0. When vui\_aspect\_ratio\_idc is equal to 0 or vui\_sar\_width is equal to 0 or vui\_sar\_height is equal to 0, the SAR is unknown or unspecified in this Specification or may be determined by other means, such as the SARI SEI message.

**vui\_overscan\_info\_present\_flag** equal to 1 specifies that the vui\_overscan\_appropriate\_flag is present. When vui\_overscan\_info\_present\_flag is equal to 0 or is not present, the preferred display method for the video signal is unknown or unspecified or specified by external means.

**vui\_overscan\_appropriate\_flag** equal to 1 indicates that the cropped decoded pictures output are suitable for display using overscan. vui\_overscan\_appropriate\_flag equal to 0 indicates that the cropped decoded pictures output contain visually important information in the entire region out to the edges of the conformance cropping window of the picture, such that the cropped decoded pictures output should not be displayed using overscan. Instead, they should be displayed using either an exact match between the display area and the conformance cropping window, or using underscan. As used in this paragraph, the term "overscan" refers to display processes in which some parts near the borders of the cropped decoded pictures are not visible in the display area. The term "underscan" describes display processes in which the entire cropped decoded pictures are visible in the display area, but they do not cover the entire display area. For display processes that neither use overscan nor underscan, the display area exactly matches the area of the cropped decoded pictures.

NOTE 2 – For example, vui\_overscan\_appropriate\_flag equal to 1 might be used for entertainment television programming or for a live view of people in a videoconference, and vui\_overscan\_appropriate\_flag equal to 0 might be used for computer screen capture or security camera content.

**vui\_colour\_description\_present\_flag** equal to 1 specifies that vui\_colour\_primaries, vui\_transfer\_characteristics, and vui\_matrix\_coeffs are present. vui\_colour\_description\_present\_flag equal to 0 specifies that vui\_colour\_primaries, vui\_transfer\_characteristics, and vui\_matrix\_coeffs are not present.

**vui\_colour\_primaries** indicates the chromaticity coordinates of the source colour primaries. Its semantics are as specified for the ColourPrimaries parameter in Rec. ITU-T H.273 | ISO/IEC 23091-2. When the vui\_colour\_primaries syntax element is not present, the value of vui\_colour\_primaries is inferred to be equal to 2 (the chromaticity is unknown or unspecified or determined by other means not specified in this Specification). Values of vui\_colour\_primaries that are identified as reserved for future use in Rec. ITU-T H.273 | ISO/IEC 23091-2 shall not be present in bitstreams conforming to this version of this Specification. Decoders shall interpret reserved values of vui\_colour\_primaries as equivalent to the value 2.

**vui\_transfer\_characteristics** indicates the transfer characteristics function of the colour representation. Its semantics are as specified for the TransferCharacteristics parameter in Rec. ITU-T H.273 | ISO/IEC 23091-2. When the vui\_transfer\_characteristics syntax element is not present, the value of vui\_transfer\_characteristics is inferred to be equal to 2 (the transfer characteristics are unknown or unspecified or determined by other means not specified in this Specification). Values of vui\_transfer\_characteristics that are identified as reserved for future use in Rec. ITU-T H.273 | ISO/IEC 23091-2 shall not be present in bitstreams conforming to this version of this Specification. Decoders shall interpret reserved values of vui\_transfer\_characteristics as equivalent to the value 2.

**vui\_matrix\_coeffs** describes the equations used in deriving luma and chroma signals from the green, blue, and red, or Y, Z, and X primaries. Its semantics are as specified for MatrixCoefficients in Rec. ITU-T H.273 | ISO/IEC 23091-2.

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

– BitDepthC is equal to BitDepthY.

– ChromaFormatIdc is equal to 3 (the 4:4:4 chroma format).

The specification of the use of vui\_matrix\_coeffs equal to 0 under all other conditions is reserved for future use by ITU‑T | ISO/IEC.

vui\_matrix\_coeffs shall not be equal to 8 unless one of the following conditions is true:

– BitDepthC is equal to BitDepthY,

– BitDepthC is equal to BitDepthY + 1 and ChromaFormatIdc is equal to 3 (the 4:4:4 chroma format).

The specification of the use of vui\_matrix\_coeffs equal to 8 under all other conditions is reserved for future use by ITU‑T | ISO/IEC.

When the vui\_matrix\_coeffs syntax element is not present, the value of vui\_matrix\_coeffs is inferred to be equal to 2 (unknown or unspecified or determined by other means not specified in this Specification).

**vui\_full\_range\_flag** indicates the scaling and offset values applied in association with the matrix coefficients. Its semantics are as specified for the VideoFullRangeFlag parameter in Rec. ITU-T H.273 | ISO/IEC 23091-2. When not present, the value of vui\_full\_range\_flag is inferred to be equal to 0.

**vui\_chroma\_loc\_info\_present\_flag** equal to 1 specifies that either vui\_chroma\_sample\_loc\_type\_frame or both vui\_chroma\_sample\_loc\_type\_top\_field and vui\_chroma\_sample\_loc\_type\_bottom\_field are present. vui\_chroma\_loc\_info\_present\_flag equal to 0 specifies that vui\_chroma\_sample\_loc\_type\_frame, vui\_chroma\_sample\_loc\_type\_top\_field, and vui\_chroma\_sample\_loc\_type\_bottom\_field are not present.

When ChromaFormatIdc is not equal to 1, vui\_chroma\_loc\_info\_present\_flag should be equal to 0.

**vui\_chroma\_sample\_loc\_type\_frame**, **vui\_chroma\_sample\_loc\_type\_top\_field**, and **vui\_chroma\_sample\_loc\_type\_bottom\_field**, when present, specify the location of chroma samples as follows:

– If GeneralProgressiveSourceFlag is equal to 1, GeneralInterlacedSourceFlag is equal to 0, and ChromaFormatIdc is equal to 1 (4:2:0 chroma format), vui\_chroma\_sample\_loc\_type\_frame specifies the location of chroma samples for both fields of each frame of the CLVS as shown in Figure 1.

– Otherwise, if ChromaFormatIdc is equal to 1 (4:2:0 chroma format), vui\_chroma\_sample\_loc\_type\_top\_field and vui\_chroma\_sample\_loc\_type\_bottom\_field specify the location of chroma samples for each top field and bottom field of the CLVS, respectively, as shown in Figure 1.

– Otherwise (ChromaFormatIdc is not equal to 1), the values of the syntax elements chroma\_sample\_loc\_type, vui\_chroma\_sample\_loc\_type\_top\_field and vui\_chroma\_sample\_loc\_type\_bottom\_field shall be ignored.

A screenshot of a computer screen

Description automatically generated

**Figure 1 – Location of chroma samples for top and bottom fields for ChromaFormatIdc equal to 1 (4:2:0 chroma format) as a function of vui\_chroma\_sample\_loc\_type\_top\_field and vui\_chroma\_sample\_loc\_type\_bottom\_field in the range of 0 to 5, inclusive**

When ChromaFormatIdc is equal to 2 (4:2:2 chroma format), the nominal positions of the chroma samples are co-sited with the corresponding luma samples and the nominal locations in a picture are as shown in Figure 2.



**Figure****2 – Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a picture**

When ChromaFormatIdc is equal to 3 (4:4:4 chroma format), the nominal positions of the chroma samples are such that all array samples are co-sited for all cases of pictures and the nominal locations in a picture are as shown in Figure 3.



**Figure****3 – Nominal vertical and horizontal locations of 4:4:4 luma and chroma samples in a picture**

When ChromaFormatIdc is equal to 0, there is no chroma sample array.

When present, the values of vui\_chroma\_sample\_loc\_type\_frame, vui\_chroma\_sample\_loc\_type\_top\_field and vui\_chroma\_sample\_loc\_type\_bottom\_field shall be in the range of 0 to 6, inclusive.

When ChromaFormatIdc is equal to 1 and vui\_chroma\_loc\_info\_present\_flag is equal to 0, vui\_chroma\_sample\_loc\_type\_frame is not present and is inferred to be equal to 6, which indicates that the the location of the chroma samples is unknown or unspecified or specified by other means not specified in this Specification. When vui\_chroma\_sample\_loc\_type\_top\_field and vui\_chroma\_sample\_loc\_type\_bottom\_field are not present, the values of vui\_chroma\_sample\_loc\_type\_top\_field and vui\_chroma\_sample\_loc\_type\_bottom\_field are inferred to be equal to vui\_chroma\_sample\_loc\_type\_frame.

NOTE 3 – In Rec. ITU-T H.266 | ISO/IEC 23090-3 and Rec. ITU-T H.265 | ISO/IEC 23008-2, a nominal chroma sampling type is identified for ChromaFormatIdc equal to 1 that corresponds to vui\_chroma\_sample\_loc\_type\_frame, vui\_chroma\_sample\_loc\_type\_top\_field and vui\_chroma\_sample\_loc\_type\_bottom\_field equal to 0.

Figure 4 illustrates the indicated relative position of the top-left chroma sample when ChromaFormatIdc is equal to 1 (i.e., the 4:2:0 chroma format), and vui\_chroma\_sample\_loc\_type\_top\_field or vui\_chroma\_sample\_loc\_type\_bottom\_field is equal to the value of a variable Chroma420LocType. The region represented by the top-left 4:2:0 chroma sample (depicted as a large grey, solid-line square with a large grey dot at its centre) is shown relative to the region represented by the top-left luma sample (depicted as a small black square with a small black dot at its centre). The regions represented by neighbouring luma samples are depicted as small grey, dotted-line squares with small grey dots at their centres.

A black and white image of a diagram

Description automatically generated

**Figure 4 – Location of the top-left chroma sample when ChromaFormatIdc is equal to 1  
(4:2:0 chroma format) and Chroma420LocType is equal to 0 to 5, inclusive, from left to right**

The relative spatial positioning of the chroma samples, as illustrated in Figure 5, can be expressed by defining two variables HorizontalOffsetC and VerticalOffsetC as a function of ChromaFormatIdc and the variable Chroma420LocType as given by Table 3, where HorizontalOffsetC is the horizontal (x) position of the centre of the top-left chroma sample relative to the centre of the top-left luma sample in units of luma samples and VerticalOffsetC is the vertical (y) position of the centre of the top-left chroma sample relative to the centre of the top-left luma sample in units of luma samples.

In a typical FIR filter design, when ChromaFormatIdc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), HorizontalOffsetC and VerticalOffsetC would serve as the phase offsets for the horizontal and vertical filter operations, respectively, for separable downsampling from 4:4:4 chroma format to the chroma format indicated by ChromaFormatIdc.

A black background with white dots and circles

Description automatically generated

**Figure 5 – Location of the top-left chroma sample when** **ChromaFormatIdc is equal to 1  
(4:2:0 chroma format) when Chroma420LocType is equal to 1**

**Table 3 – Definition of HorizontalOffsetC and VerticalOffsetC  
as a function of ChromaFormatIdc and Chroma420LocType**

|  |  |  |  |
| --- | --- | --- | --- |
| **ChromaFormatIdc** | **Chroma420LocType** | **HorizontalOffsetC** | **VerticalOffsetC** |
| 1 (4:2:0) | 0 | 0 | 0.5 |
| 1 (4:2:0) | 1 | 0.5 | 0.5 |
| 1 (4:2:0) | 2 | 0 | 0 |
| 1 (4:2:0) | 3 | 0.5 | 0 |
| 1 (4:2:0) | 4 | 0 | 1 |
| 1 (4:2:0) | 5 | 0.5 | 1 |
| 2 (4:2:2) | – | 0 | 0 |
| 3 (4:4:4) | – | 0 | 0 |

When ChromaFormatIdc is equal to 1 (4:2:0 chroma format) and the decoded video content is intended for interpretation according to Rec. ITU-R BT.2020 or Rec. ITU-R BT.2100, vui\_chroma\_loc\_info\_present\_flag should be equal to 1, and vui\_chroma\_sample\_loc\_type\_frame, vui\_chroma\_sample\_loc\_type\_top\_field, and vui\_chroma\_sample\_‌loc\_‌type\_‌bottom\_‌field (as applicable) should be equal to 2.

**vui\_modality\_info\_present\_flag** equal to 1 specifies that vui\_modality\_type is present in the VUI parameters. vui\_modality\_info\_present\_flagequal to 0 specifies that vui\_modality\_type is not present in the VUI parameters.

**vui\_modality\_type** indicates the type of modality of the decoded picture as specified in Table 4. When not present, the value of vui\_modality\_type is inferred to be equal to 0, denoting that the modality type of the picture is unknown or unspecified or determined by other means not specified in this Specification.

**Table 4– Mapping of vui\_modality\_type to the type of picture modalities**

|  |  |
| --- | --- |
| **vui\_modality\_type** | **type of picture modality** |
| 0 | Unspecified |
| 1 | Visible Picture |
| 2 | Infrared Picture |
| 3 | Thermal Infrared Picture |
| 4..255 | Reserved for future use |

**vui\_payload\_bit\_equal\_to\_one** shall be equal to 1.

**vui\_payload\_bit\_equal\_to\_zero** shall be equal to 0.

*Changes to VSEI subclause 8.28 for adding the support of grouping of NNPFs*

Modify text of JVET-AE2032 subclause 8.28 per JVET-AF0050, JVET-AF0051, JVET-AF0091, JVET-AF0093, and JVET-AF0094 as follows

8.28 Neural-network post-filter SEI messages

**8.28.1 General** **post-processing filtering process using NNPFs**

**8.28.1.1 General**

Input to this process is a bitstream BitstreamToFilter. Output of this process is a list of NNPF output pictures ListNnpfOutputPics.

First, BitstreamToFilter is decoded, and the list CroppedDecodedPictures is set to be the list of the cropped decoded pictures in output order resulted from decoding BitstreamToFilter.

Second, NnpfCand is set to contain any single NNPF or any single NNPF group. When NnpfCand contains an NNPF group with nnpfgc\_grouping\_type equal to 3, the subseequnt specifications of this subclause apply when NnpfCand is set to contain individually each member NNPF, if any, and each member NNPF group, if any, of the NNPF group with nnpfgc\_grouping\_type equal to 3.

Third, the filtering process for one picture, as specified in subclause 8.28.1.2, is repeatedly invoked, in output order, for each cropped decoded picture that is in CroppedDecodedPictures and for which the single NNPF contained in NnpfCand or the single NNPF group contained in NnpfCand, or one or more NNPFs or NNPF groups defined as alternatives or alternating in the NNPF group contained in NnpfCand are activated.

The order of the pictures in ListNnpfOutputPics is in output order.

Within ListNnpfOutputPics there shall be no more than one picture pertaining to any particular output time instance. [Ed. Check phrasing of this. “Pertains” is not used in a similar way anywhere in the standard.] When for any particular picture in CroppedDecodedPictures there are multiple NNPFs activated and only one of the NNPFs is allowed to be chosen to be applied although any of the NNPFs may be chosen, the above constraint shall apply regardless of which NNPF is chosen to be applied to the particular picture.

BitstreamToFilter may be processed multiple times to generate multiple different ListNnpfOutputPics through the second and third steps above.

**8.28.1.2 Filtering process for one picture**

The filtering process specified in this subclause applies to each cropped decoded picture, referred to as the current picture, that is in CroppedDecodedPictures and for which one or more NNPFs or NNPF groups in NnpfCand are activated.

An NNPF or an NNPF group to be applied to the current picture is selected as follows:

– If NnpfCand contains a single NNPF and that NNPF is activated for the current picture according to an NNPFA SEI message, that NNPF is selected to be applied to the current picture.

– Otherwise, if NnpfCand contains an NNPF group with nnpfgc\_grouping\_type equal to 2 and any NNPF of the NNPF group is activated for the current picture according to NNPFA SEI message, that NNPF is selected to be applied to the current picture.

– Otherwise, if NnpfCand contains an NNPF group with nnpfgc\_grouping\_type equal to 0 and that NNPF group is activated for the current picture according to an NNPFGA SEI message, that NNPF group is selected to be applied to the current picture.

– Otherwise (NnpfCand contains an NNPF group with nnpfgc\_grouping\_type equal to 1), the following applies:

– A set of candidate NNPFs or NNPF groups candSet is initially empty and then set to contain the following:

– The NNPFs that are activated for the current picture according to NNPFA SEI messages and are included in the NNPF group contained in NnpfCand.

– The NNPF groups that are activated for the current picture according to NNPFGA SEI messages and are included in the NNPF group contained in NnpfCand.

– For each candidate NNPF or NNPF group candFilter in candSet, the following applies:

– When one or more of the input pictures of candFilter are input pictures to the NNPF or NNPF group prevFilter that was used in any previous invocation of the filtering process specified in this subclause for the same NnpfCand, candFilter is excluded from candSet.

– Any NNPF or NNPF group remaining in candSet is selected to be applied to the current picture.

When applying an NNPF to the current picture, the following applies:

– The filtered and/or interpolated pictures are generated by the NNPF by applying the NNPF process specified in the semantics of the NNPFC SEI message, in a patch-wise manner, to the current picture.

– The order of the pictures generated by the NNPF by applying the NNPF process being stored into the output tensor of the NNPF is in output order.

– The pictures generated by the NNPF and output by the NNPF process are included into ListNnpfOutputPics, in the same order as when the pictures are stored into the output tensor of the NNPF.

When applying an NNPF group to the current picture, the following applies:

– The filtered and/or interpolated pictures are generated by applying the NNPF process specified in the semantics of the NNPFC SEI message, in a patch-wise manner, as specified in the semantics of the NNPFGA SEI message activating the NNPF group.

– The pictures in FinalNnpfgaOutputPicList are included into ListNnpfOutputPics, in the same order as the pictures are stored in FinalNnpfgaOutputPicList.

**8.28.2 Neural-network post-filter characteristics SEI message**

8.28.2.1 Neural-network post-filter characteristics SEI message syntax

|  |  |
| --- | --- |
| nn\_post\_filter\_characteristics( payloadSize ) { | **Descriptor** |
| … | u(1) |
| if( nnpfc\_complexity\_info\_present\_flag ) { |  |
| **nnpfc\_parameter\_type\_idc** | u(2) |
| if( nnpfc\_parameter\_type\_idc != 2 ) |  |
| **nnpfc\_log2\_parameter\_bit\_length\_minus3** | u(2) |
| **nnpfc\_num\_parameters\_idc** | u(6) |
| **nnpfc\_num\_kmac\_operations\_idc** | ue(v) |
| **nnpfc\_total\_kilobyte\_size** | ue(v) |
| } |  |
| **nnpfc\_metadata\_extension\_num\_bits** | ue(v) |
| if( nnpfc\_metadata\_extension\_num\_bits > 0 ) { |  |
| if( nnpfc\_purpose = = 0 ) { |  |
| **nnpfc\_application\_purpose\_tag\_uri\_present\_flag** | u(1) |
| if( nnpfc\_application\_purpose\_tag\_uri\_present\_flag ) |  |
| **nnpfc\_application\_purpose\_tag\_uri** | st(v) |
| } |  |
| **nnpfc\_reserved\_metadata\_extension** /\*Remaining bits of the metadata extension\*/ | u(v) |
| } |  |
| } |  |
| … |  |
| } |  |

8.28.2.2 Neural-network post-filter characteristics SEI message semantics

…

**nnpfc\_purpose** indicates the purpose of the NNPF as specified in Table 20, where ( nnpfc\_purpose & bitMask ) not equal to 0 indicates that the NNPF has the purpose associated with the bitMask value in Table 20. When nnpfc\_purpose is greater than 0 and ( nnpfc\_purpose & bitMask ) is equal to 0, the purpose associated with the bitMask value is not applicable to the NNPF. When nnpfc\_pupose is equal to 0, the NNPF may be used as determined by the application and as specified by the nnpfc\_application\_purpose\_tag\_uri.

The value of nnpfc\_purpose shall be in the range of 0 to 63, inclusive, in bitstreams conforming to this edition of this document. Values of 64 to 65 535, inclusive, for nnpfc\_purpose are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this edition of this document. Decoders conforming to this edition of this document shall ignore NNPFC SEI messages with nnpfc\_purpose in the range of 64 to 65 535, inclusive.

**Table 20 – Definition of nnpfc\_purpose**

|  |  |
| --- | --- |
| **bitMask** | **Interpretation** |
| 0x01 | General visual quality improvement |
| 0x02 | Chroma upsampling (from the 4:2:0 chroma format to the 4:2:2 or 4:4:4 chroma format, or from the 4:2:2 chroma format to the 4:4:4 chroma format) |
| 0x04 | Resolution resampling (increasing or decreasing the width or height) |
| 0x08 | Picture rate upsampling |
| 0x10 | Bit depth upsampling (increasing the luma bit depth or the chroma bit depth) |
| 0x20 | Colourization |

The variables chromaUpsamplingFlag, resolutionResamplingFlag, pictureRateUpsamplingFlag, bitDepthUpsamplingFlag, and colourizationFlag, specifying whether nnpfc\_purpose indicates the purpose of the NNPF to include chroma upsampling, resolution resampling, picture rate upsampling, bit depth upsampling, and colourization, respectively, are derived as follows:

chromaUpsamplingFlag = ( ( nnpfc\_purpose & 0x02 ) > 0 ) ? 1 : 0  
resolutionResamplingFlag = ( ( nnpfc\_purpose & 0x04 ) > 0 ) ? 1 : 0  
pictureRateUpsamplingFlag = ( ( nnpfc\_purpose & 0x08 ) > 0 ) ? 1 : 0 (76)  
bitDepthUpsamplingFlag = ( ( nnpfc\_purpose & 0x10 ) > 0 ) ? 1 : 0  
colourizationFlag = ( ( nnpfc\_purpose & 0x20 ) > 0 ) ? 1 : 0

NOTE 2– When a reserved value of nnpfc\_purpose is taken into use in the future by ITU-T | ISO/IEC, the syntax of this SEI message could be extended with syntax elements whose presence is conditioned by nnpfc\_purpose being equal to that value.

When ChromaFormatIdc is equal to 3, chromaUpsamplingFlag shall be equal to 0.

When ChromaFormatIdc or chromaUpsamplingFlag is not equal to 0, colourizationFlag shall be equal to 0.

When pictureRateUpsamplingFlag is equal to 1 and the input picture with index 0 is associated with a frame packing arrangement SEI message with fp\_arrangement\_type equal to 5, all input pictures are associated with a frame packing arrangement SEI message with fp\_arrangement\_type equal to 5 and the same value of fp\_current\_frame\_is\_frame0\_flag.

…

**nnpfc\_application\_purpose\_tag\_uri\_present\_flag** equal to 1indicates that the nnpfc\_application\_purpose\_tag\_uri syntax element is present in this NNPFC SEI message. nnpfc\_application\_purpose\_tag\_uri\_present\_flag equal to 0indicates that the nnpfc\_application\_purpose\_tag\_uri syntax element is not present in this NNPFC SEI message. When not present nnpfc\_application\_purpose\_tag\_uri\_present\_flag is inferred to be equal to 0.

**nnpfc\_application\_purpose\_tag\_uri** specifies a tag URI with syntax and semantics as specified in IETF RFC 4151 identifying the application determined purpose of the NNPF, when nnpfc\_purpose is equal to 0.

NOTE 4 – nnpfc\_application\_purpose\_tag\_uri enables uniquely identifying the application determined purpose of NNPF without needing a central registration authority.

…

**nnpfc\_metadata\_extension\_num\_bits** equal to 0 specifies that nnpfc\_reserved\_metadata\_extension is not present. nnpfc\_metadata\_extension\_num\_bits greater than 0 specifies the length, in bits, of nnpfc\_reserved\_metadata\_extension. nnpfc\_metadata\_extension\_num\_bits shall be in the range of 0 to 2048 in this edition of this document when nnpfc\_purpose is not equal to 0 and in the range of 1 to 2048 when nnpfc\_purpose is equal to 0. Values in the range of 2049 to 4096, inclusive, for nnpfc\_metadata\_extension\_num\_bits are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this edition of this document. Decoders conforming to this edition of this document shall allow any value of nnpfc\_metadata\_extension\_num\_bits in the range of 0 to 4096, inclusive. Values of nnpfc\_metadata\_extension\_num\_bits greater than 4096 shall not be present in bitstreams conforming to this edition of this document and are not reserved for future use.

**nnpfc\_reserved\_metadata\_extension** shall not be present in bitstreams conforming to this edition of this document. However, decoders conforming to this edition of this document shall ignore the presence and value of nnpfc\_reserved\_metadata\_extension. When present, and when nnpfc\_purpose is equal to 0 and nnpfc\_application\_purpose\_tag\_uri\_present\_flag is equal to 1 the length, in bits, of nnpfc\_reserved\_metadata\_extension is equal to nnpfc\_metadata\_extension\_num\_bits – Length of (nnpfc\_application\_purpose\_tag\_uri) – 1. When present and when nnpfc\_purpose is equal to 0 and nnpfc\_application\_purpose\_tag\_uri\_present\_flag is equal to 0 the length, in bits, of nnpfc\_reserved\_metadata\_extension is equal to nnpfc\_metadata\_extension\_num – 1 bits. When present and when nnpfc\_purpose is not equal to 0 the length, in bits, of nnpfc\_reserved\_metadata\_extension is equal to nnpfc\_metadata\_extension\_num\_bits.

* + 1. **Neural-network post-filter group characteristics SEI message**
       1. **Neural-network post-filter group characteristics SEI message syntax**

|  |  |
| --- | --- |
| nn\_post\_filter\_group\_characteristics( payloadSize ) { | **Descriptor** |
| **nnpfgc\_id** | ue(v) |
| **nnpfgc\_grouping\_type** | ue(v) |
| if( nnpfgc\_grouping\_type = = 0 | | nnpfgc\_grouping\_type = = 2 ) |  |
| **nnpfgc\_purpose** | u(16) |
| **nnpfgc\_num\_members\_minus2** | ue(v) |
| for( i = 0; i <= nnpfgc\_num\_members\_minus2 + 1; i++ ) { |  |
| **nnpfgc\_member\_id**[ i ] | ue(v) |
| if( nnpfgc\_grouping\_type = = 1 | | nnpfgc\_grouping\_type = = 3 ) |  |
| **nnpfgc\_member\_purpose**[ i ] | u(16) |
| } |  |
| **nnpfgc\_complexity\_info\_present\_flag** | u(1) |
| if( nnpfgc\_complexity\_info\_present\_flag ) { |  |
| **nnpfgc\_parameter\_type\_idc** | u(2) |
| if( nnpfgc\_parameter\_type\_idc != 2 ) |  |
| **nnpfgc\_log2\_parameter\_bit\_length\_minus3** | u(2) |
| **nnpfgc\_num\_parameters\_idc** | u(6) |
| **nnpfgc\_num\_kmac\_operations\_idc** | ue(v) |
| **nnpfgc\_total\_kilobyte\_size** | ue(v) |
| } |  |
| } |  |

* + - 1. **Neural-network post-filter group characteristics SEI message semantics**

The neural-network post-filter group characteristics (NNPFGC) SEI message specifies a neural network post-filter (NNPF) group. It is indicated by the SEI message if the NNPF group defines an NNPF cascade or defines NNPFs or NNPF groups of NNPF cascades that are alternatives to each other. The use of NNPF groups of NNPF cascades for specific pictures is indicated with neural-network post-filter group activation (NNPFGA) SEI messages. An NNPF group may have members that are NNPF or NNPF group. An NNPF is a direct member of an NNPF group when it is included in the list of members in the NNPFGC SEI message of the NNPF group. An NNPF is an indirect member of an NNPF group nnpfg\_1 when it is a member of another NNPF group nnpfg\_2 and nnpfg\_2 is a member of nnpfg\_1.

**nnpfgc\_id** contains an identifying number that may be used to identify an NNPF group. The value of nnpfgc\_id shall be in the range of 0 to 232 − 2, inclusive. Values of nnpfgc\_id from 256 to 511, inclusive, and from 231 to 232 − 2, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders conforming to this edition of this document encountering an NNPFGC SEI message with nnpfgc\_id in the range of 256 to 511, inclusive, or in the range of 231 to 232 − 2, inclusive, shall ignore the SEI message. The value of nnpfgc\_id shall not be equal to any nnpfc\_id value of any NNPFC SEI message present in the same CLVS. When the value of nnpfgc\_id of an NNPFGC SEI message nnpfgcSeiA is equal to the value of nnpfgc\_id of another NNPFGC SEI message nnpfgcSeiB present in the same CLVS, nnpfgcSeiA and nnpfgcSeiB shall be identical.

**nnpfgc\_grouping\_type** equal to 0 indicates that this SEI message specifies a group of cascaded neural-network post-filters.

nnpfgc\_grouping\_type equal to 1 indicates that the NNPFs or NNPF groups identified by the nnpfgc\_member\_id[ i ] are alternatives to each other out of which the post-processor should select only one to be applied.

nnpfgc\_grouping\_type equal to 2 indicates that this SEI message specifies a group of NNPFs that are intended to be used jointly and are activated in an alternating manner so that at most one of these NNPFs is activate for any picture.

nnpfgc\_grouping\_type equal to 3 indicates that the NNPFs or NNPF groups identified by the nnpfgc\_member\_id[ i ] are intended to be used in parallel.

nnpfgc\_grouping\_type equal to 4 indicates that the NNPFs or NNPF groups identified by the nnpfgc\_member\_id[ i ] are optional, i.e., may or may not be applied by the post-processor.

The value of nnpfgc\_grouping\_type shall be in the range of 0 to 255, inclusive. Values of nnpfgc\_grouping\_type in the range of 5 to 255, inclusive, are reserved for future specification by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this edition of this document. Decoders conforming to this edition of this document shall ignore NNPFGC SEI messages with nnpfgc\_grouping\_type in the range of 5 to 255, inclusive.

**nnpfgc\_purpose** has the semantics of nnpfc\_purpose but with the exception that the semantics are specified for the NNPF group defined by this SEI message rather than the NNPF defined by an NNPFC SEI message. When present, the value of nnpfgc\_purpose shall be the union of the values of nnpfc\_purpose of direct and indirect members of the NNPF group.

**nnpfgc\_num\_members\_minus2** plus 2indicates the number of NNPFs or NNPF groups in the NNPF group that this SEI message defines.

**nnpfgc\_member\_id**[ i ] indicates the i-th member in the NNPF group defined by this SEI message as follows:

– If there is an NNPF with nnpfc\_id equal to nnpfgc\_member\_id[ i ] defined in the CLVS, the i-th member in the NNPF group defined by this SEI message is an NNPF that has nnpfc\_id equal to nnpfgc\_member\_id[ i ].

– Otherwise (there is no NNPF with nnpfc\_id equal to nnpfgc\_member\_id[ i ] defined in the CLVS), the i-th member in the NNPF group defined by this SEI message is an NNPF group with nnpfgc\_id equal to nnpfgc\_member\_id[ i ].

When an nnpfgc\_member\_id[ i ] value references an nnpfgc\_id value of an NNPFGC SEI message nnpfgcSei, it is a requirement of bitstream conformance that the NNPFGC SEI message nnpfgcSei shall have nnpfgc\_grouping\_type equal to 0.

NOTE 1 – In other words, when a second NNPF group is a member of a first NNPF group, the type of the second NNPF group cannot be any other than a cascade of NNPFs.

When nnpfgc\_grouping\_type is equal to 0 or 2, it is a requirement of bitstream conformance that there is an NNPF with nnpfc\_id value equal to nnpfgc\_member\_id[ i ] defined in the CLVS.

NOTE 2 – In other words, the members of a cascade or alternating NNPF group are individual NNPFs and cannot be NNPF groups.

When nnpfgc\_grouping\_type is equal to 1, 3, or 4, it is a requirement of bitstream conformance that there is an NNPF with nnpfc\_id value equal to nnpfgc\_member\_id[ i ] or an NNPF group with nnpfgc\_id value equal to nnpfgc\_member\_id[ i ] defined in the CLVS.

When nnpfgc\_grouping\_type is equal to 0, the NNPFs with nnpfc\_id equal to nnpfgc\_member\_id[ i ] are performed in cascade in increasing order of i, as activated by an NNPFGA SEI message with nnpfga\_target\_id equal to nnpfgc\_id.

**nnpfgc\_member\_purpose**[ i ] has the semantics of nnpfc\_purpose but with the exception that the semantics are specified for the NNPFC or NNPFGC associated with nnpfgc\_member[ i ]. When present, the value of nnpfgc\_member\_purpose[ i ] shall be as follows:

– If nnpfgc\_member\_id[ i ] is equal to the value of nnpfc\_id of an NNPF, the value of nnpfgc\_member\_purpose[ i ] shall be equal to the value of nnpfc\_purpose of the NNPF.

– Otherwise (nnpfgc\_member\_id[ i ] is equal to the value of nnpfgc\_id of an NNPF group ), the value of nnpfgc\_member\_purpose[ i ] shall be equal to nnpfgc\_purpose of the associatied NNPF group.

nnpfgc\_complexity\_info\_present\_flag, nnpfgc\_parameter\_type\_idc, nnpfgc\_log2\_parameter\_bit\_length\_minus3, nnpfgc\_num\_parameters\_idc, nnpfgc\_num\_kmac\_operations\_idc, and nnpfgc\_total\_kilobyte\_size have the semantics of nnpfc\_complexity\_info\_present\_flag, nnpfc\_parameter\_type\_idc, nnpfc\_log2\_parameter\_bit\_length\_minus3, nnpfc\_num\_parameters\_idc, nnpfc\_num\_kmac\_operations\_idc, and nnpfc\_total\_kilobyte\_size, respectively, but with the exception that the semantics are specified for the NNPF group defined by this SEI message rather than the NNPF defined by an NNPFC SEI message. When nnpfgc\_grouping\_type is equal to 1, nnpfgc\_complexity\_info\_present\_flag shall be equal to 0.

* + 1. **Neural-network post-filter group activation SEI message**
       1. **Neural-network post-filter group activation SEI message syntax**

|  |  |
| --- | --- |
| nn\_post\_filter\_group\_activation( payloadSize ) { | **Descriptor** |
| **nnpfga\_target\_id** | ue(v) |
| **nnpfga\_cancel\_flag** | u(1) |
| if( !nnpfga\_cancel\_flag ) { |  |
| **nnpfga\_persistence\_flag** | u(1) |
| **nnpfga\_no\_prev\_clvs\_flag** | u(1) |
| if( nnpfga\_persistence\_flag ) |  |
| **nnpfga\_no\_foll\_clvs\_flag** | u(1) |
| **nnpfga\_num\_filters\_minus2** | ue(v) |
| for( i = 0; i <= nnpfga\_num\_filters\_minus2 + 1; i++ ) { |  |
| **nnpfga\_target\_base\_flag**[ i ] | u(1) |
| **nnpfga\_input\_all\_pics\_flag**[ i ] | u(1) |
| if( !nnpfga\_input\_all\_pics\_flag[ i ] ) { |  |
| **nnpfga\_num\_input\_pics\_minus1**[ i ] | ue(v) |
| for( j = 0; j <= nnpfga\_num\_input\_pics\_minus1[ i ]; j++ ) |  |
| **nnpfga\_input\_pic\_skip\_count**[ i ][ j ] | ue(v) |
| } |  |
| **nnpfga\_num\_output\_entries**[ i ] | ue(v) |
| for( j = 0; j < nnpfga\_num\_output\_entries[ i ]; j++ ) |  |
| **nnpfga\_output\_flag**[ i ][ j ] | u(1) |
| } |  |
| **nnpfga\_num\_output\_pic\_update** | ue(v) |
| for( i = 0; i < nnpfga\_num\_output\_pic\_update; i++ ) |  |
| **nnpfga\_output\_pic\_update\_flag**[ i ] | u(1) |
| } |  |
| } |  |

* + - 1. **Neural-network post-filter group activation SEI message semantics**

The neural-network post-filter group activation (NNPFGA) SEI message activates or de-activates the possible use of the target neural-network post-processing filter group (NNPFG) of NNPF groups, identified by nnpfga\_target\_id, for post-processing filtering of a set of pictures.

nnpfgc\_grouping\_type for the identfied NNPF group, which has nnpfgc\_id equal to nnpfga\_target\_id, shall be equal to 0 (cascade) or 1 (alternatives). When nnpfgc\_grouping\_type of the identified NNPF group is equal to 1, each member of the group shall have the same number of input pictures and NNPF output pictures.

For a particular picture for which the NNPFG is activated, the associated NNPFGC SEI message is the last NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id, that precedes the first VCL NAL unit of the current picture in decoding order. If the nnpfgc\_grouping\_type in the associated NNPGC SEI message is equal to 0, the target NNPFG is the NNPFG specified by the associated NNPFGC SEI message. Otherwise (nnpfgc\_grouping\_type in the associated NNPFGC SEI message is equal to 1), the target NNPFG is any NNPFG, if any, that is a member of the NNPFG specified by the associated NNPFGC SEI message. The NNPFs of the target NNPFG are defined by the NNPFC SEI messages that have nnpfc\_id equal to any nnpfgc\_member\_id[ i ] value of the target NNPFG and are present in the current picture unit or precede the current picture in decoding order.

NOTE 1 – The members of the target NNPFG are individual NNPFs and cannot be NNPF groups.

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

– Input picture width and height in units of luma samples, denoted herein by InitCroppedWidth[ idx ] and InitCroppedHeight[ idx ], respectively, of the candidate input pictures with index idx in the range of 0 to numCandInputPics − 1, inclusive, that may be used as input for the NNPFG.

– Luma sample array InitCroppedYPic[ idx ] and chroma sample arrays InitCroppedCbPic[ idx ] and InitCroppedCrPic[ idx ], when present, of the candidate input pictures with index idx in the range of 0 to numCandInputPics − 1, inclusive, that may be used as input for the NNPFG.

– Bit depth BitDepthY for the luma sample array of the candidate input pictures.

– Bit depth BitDepthC for the chroma sample arrays, if any, of the candidate input pictures.

– A chroma format indicator, denoted herein by ChromaFormatIdc, as described in subclause ‎7.3.

– When nnpfc\_auxiliary\_inp\_idc is equal to 1, a filtering strength control value array StrengthControlVal[ idx ] that shall contain real numbers in the range of 0 to 1, inclusive, of the candidate input pictures with index idx in the range of 0 to numCandInputPics − 1, inclusive.

Candidate input picture with index 0 corresponds to the picture for which the NNPFG is activated by this NNPFGA SEI message. Candidate input picture with index i in the range of 1 to numCandInputPics − 1, inclusive, precedes the candidate input picture with index i − 1 in output order. Let candInputPicList[ 0 ] be the list of candidate input pictures in inverse output order.

**nnpfga\_target\_id** indicates the target NNPFG, which is specified by the NNPFGC SEI message that pertains to the current picture and have nnpfgc\_id equal to nnpfga\_target\_id. [Ed. Check phrasing of this. “Pertains” is not used in a similar way anywhere in the standard.]

The value of nnpfga\_target\_id shall be in the range of 0 to 232 − 2, inclusive.

An NNPFGA SEI message with a particular value of nnpfga\_target\_id shall not be present in a current PU unless there is an NNPFGC SEI message with nnpfgc\_id equal to the particular value of nnpfga\_target\_id and nnpfgc\_grouping\_type equal to 0 present in the current PU or in a PU that precedes the current PU in decoding order within the current CLVS.

When a PU contains both an NNPFGC SEI message with a particular value of nnpfgc\_id and an NNPFGA SEI message with nnpfga\_target\_id equal to the particular value of nnpfgc\_id, the NNPFGC SEI message shall precede the NNPFGA SEI message in decoding order.

**nnpfga\_cancel\_flag** equal to 1 indicates that the persistence of the target NNPFG established by any previous NNPFGA SEI message with the same nnpfga\_target\_id as the current SEI message is cancelled, i.e., the target NNPFG is no longer used unless it is activated by another NNPFGA SEI message with the same nnpfga\_target\_id as the current SEI message and nnpfga\_cancel\_flag equal to 0. nnpfga\_cancel\_flag equal to 0 indicates that the target NNPFG is activated for use.

**nnpfga\_persistence\_flag** specifies the persistence of the target NNPFG for the current layer.

nnpfga\_persistence\_flag equal to 0 specifies that the target NNPFG may be used for post-processing filtering for the current picture only.

nnpfga\_persistence\_flag equal to 1 specifies that the target NNPFG may be used for post-processing filtering for the current picture and 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 associated with a NNPFGA SEI message with the same nnpfga\_target\_id as the current SEI message that follows the current picture in output order.

NOTE 2 – The target NNPFG is not applied for this subsequent picture in the current layer associated with a NNPFGA SEI message with the same nnpfga\_target\_id as the current SEI message.

Let the nnpfgcTargetPictures be the set of pictures to which the last NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id that precedes the current NNPFGA SEI message in decoding order pertains. [Ed. Check phrasing of this. “Pertains” is not used in a similar way anywhere in the standard.] Let nnpfgaTargetPictures be the set of pictures for which the target NNPFG is activated by the current NNPFGA SEI message. It is a requirement of bitstream conformance that any picture included in nnpfgaTargetPictures shall also be included in nnpfgcTargetPictures.

**nnpfga\_no\_prev\_clvs\_flag** equal to 1 specifies that the input pictures for the NNPF group do not originate from a previous CLVS. nnpfga\_no\_prev\_clvs\_flag equal to 0 specifies that the input pictures for the NNPF group may or may not originate from a previous CLVS.

NOTE 3 – The value of nnpfga\_no\_prev\_clvs\_flag can be changed from 0 to 1, when the current CLVS is spliced from another bitstream next to the previous CLVS and this NNPFGA SEI message would cause one or more input pictures to be selected from one or more previous CLVSs and therefore is likely to impact the output of the target NNPF group negatively.

**nnpfga\_no\_foll\_clvs\_flag** equal to 1 specifies that when this NNPFGA SEI message persists for the last PU of a CLVS in output order, the NNPFGA SEI message is treated like it persisted for the last PU, in output order, of the current layer within the bitstream. When this NNPFGA SEI message does not persist for the last PU, in output order, of a CLVS in output order or nnpfga\_no\_foll\_clvs\_flag is equal to 0, the value of nnpfga\_no\_foll\_clvs\_flag causes no specific impact.

NOTE 4 – The value of nnpfga\_no\_foll\_clvs\_flag can be changed from 0 to 1 for an NNPF group including a picture-rate-upsampling NNPF, when the following CLVS is spliced from a different bitstream next to the current CLVS. Consequently, the NNPF process of the picture-rate-upsampling NNPF interpolates pictures up to the end of the current CLVS using input pictures originating from the current CLVS only.

**nnpfga\_num\_filters\_minus2** plus 2indicates the number of NNPFs in the NNPFG that this SEI message activates. The value of nnpfga\_num\_filters\_minus2 shall be equal to the value of nnpfgc\_num\_members\_minus2 in an NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id.

**nnpfga\_target\_base\_flag**[ i ] equal to 1 specifies that the i-th NNPF in the target NNPFG is the base NNPF with nnpfc\_id equal to nnpfgc\_member\_id[ i ] in an NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id. nnpfga\_target\_base\_flag[ i ] equal to 0 specifies that the i-th NNPF in the target NNPFG is the NNPF specified by the last NNPFC SEI message that has nnpfc\_id equal to nnpfgc\_member\_id[ i ] in an NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id, precedes the first VCL NAL unit of the current picture in decoding order, and is not a repetition of the NNPFC SEI message that contains the base NNPF.

**nnpfga\_input\_all\_pics\_flag**[ i ] equal to 1 specifies that the input pictures to the i-th NNPF are selected from the list of candidate input pictures candInputPicList[ i ] without skipping. nnpfga\_input\_all\_pics\_flag[ i ] equal to 0 specifies that the input pictures to the i-th NNPF are selected from the list of candidate input pictures candInputPicList[ i ] in a manner that some candidate input pictures are skipped.

**nnpfga\_num\_input\_pics\_minus1**[ i ] specifies the number of input pictures for the i-th NNPF in the target NNPFG. When present, nnpfga\_num\_input\_pics\_minus1[ i ] shall be equal to nnpfc\_num\_input\_pics\_minus1 for an NNPF with nnpfc\_id equal to nnpfgc\_member\_id[ i ] of an NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id. When not present, nnpfga\_num\_input\_pics\_minus1[ i ] is inferred to be equal to nnpfc\_num\_input\_pics\_minus1 for an NNPF with nnpfc\_id equal to nnpfgc\_member\_id[ i ] in an NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id.

**nnpfga\_input\_pic\_skip\_count**[ i ][ j ] specifies a j-th picture count that is skipped in the list of candidate input pictures candInputPicList[ i ] when selecting input pictures for the NNPF activated by the i-th loop entry. When nnpfga\_input\_pic\_skip\_count[ i ][ j ] is not present, it is inferred to be equal to 0 for all values of j in the range of 0 to nnpfga\_num\_input\_pics\_minus1[ i ], inclusive. The variable numCandInputPics, which indicates the number of candidate input pictures to the NNPFG, is derived as follows:

numCandInputPics = 0  
for( j = 0; j <= nnpfga\_num\_input\_pics\_minus1[ 0 ]; j++ )  
 numCandInputPics += 1 + nnpfga\_input\_pic\_skip\_count[ 0 ][ j ] (xx)

Let candInputPicList[ m ] for m in the range of 1 to nnpfga\_num\_filters\_minus2 + 1, inclusive, be a list of pictures in inverse output order that is initially empty and formed in decreasing order of n in the range of 0 to m − 1, inclusive, by including each picture that is output by the NNPF process of the n-th loop entry that has no corresponding picture already present in candInputPicList[ m ], and lastly including each picture present in candInputPicList[ 0 ] that has no corresponding picture already present in candInputPicList[ m ].

When a candidate input picture candInputPicList[ m ][ idx ] for any value of m in the range of 1 to nnpfga\_num\_filters\_minus2 + 1, inclusive, is an NNPF output picture of the n-th NNPF process with the value of n being less than the value of m, the width and height of the candidate input picture are respectively equal to nnpfcOutputPicWidth and nnpfcOutputPicHeight of the NNPF output picture.

The list of input pictures inputPicList[ m ] to the NNPF of the m-th loop entry is derived as follows:

candIdx = 0  
for( k = 0; k <= nnpfga\_num\_input\_pics\_minus1[ m ]; k++ ) {  
 candIdx += nnpfga\_input\_pic\_skip\_count[ m ][ k ]  
 inputPicList[ m ][ k ] = candInputPicList[ m ][ candIdx ] (xx)  
 candIdx++  
}

It is a requirement of bitstream conformance that candIdx shall not exceed the number of pictures in candInputPicList[ m ].

It is a requirement of bitstream conformance that the pictures present in inputPicList[ m ], for any value of m in the range of 1 to nnpfga\_num\_filters\_minus2 + 1, inclusive, shall have the same width, height, bit depth, and chroma format.

For purposes of interpretation of the NNPFC SEI message with nnpfc\_id equal to nnpfgc\_member\_id[ i ] in an NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id, the following variables are specified for the i-th loop entry:

– The variables BitDepthY, BitDepthC, and ChromaFormatIdc are used as provided for the interpretation of this SEI message.

– CroppedWidth and CroppedHeight are set equal to the width and height of the pictures in inputPicList[ i ], respectively, in units of luma samples.

– For each input picture k in the range of 0 to nnpfga\_num\_input\_pics\_minus1[ i ], inclusive, the following applies:

– CroppedYPic[ k ], CroppedCbPic[ k ], and CroppedCrPic[ k ], when present, are set equal to respective sample array of inputPicList[ i ][ k ]

– When nnpfc\_auxiliary\_inp\_idc is equal to 1 for the NNPF with nnpfc\_id equal to nnpfgc\_member\_id[ i ] in an NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id, the following applies:

– It is a requirement of bitstream conformance that inputPicList[ i ][ k ] is the same as candInputPicList[ 0 ][ idx ] for any value of idx in the range of 0 to numCandInputPics − 1, inclusive.

– StrengthControlVal[ k ] is set equal to InitStrengthControlVal[ idx ].

**nnpfga\_num\_output\_entries**[ i ] specifies the number of nnpfga\_output\_flag[ i ][ j ] syntax elements present in the NNPFGA SEI message. The value of nnpfga\_num\_output\_entries[ i ] shall be in the range of 0 to NumInpPicsInOutputTensor, inclusive, for an NNPF with nnpfc\_id equal to nnpfgc\_member\_id[ i ] of an NNPFGC SEI message with nnpfgc\_id equal to nnpfga\_target\_id..

**nnpfga\_output\_flag**[ i ][ j ] equal to 1 specifies that the NNPF-generated picture that corresponds to the input picture having index InpIdx[ j ] derived for the i-th NNPF of the target NNPFG is output by the NNPF process activated by this loop entry, where the NNPF process is specified in the semantics of the NNPFC SEI message. nnpfga\_output\_flag[ i ][ j ] equal to 0 specifies that the NNPF-generated picture that corresponds to the input picture having index InpIdx[ j ] derived for the i-th NNPF of the target NNPFG is not output by the NNPF process activated by this loop entry. When nnpfga\_num\_output\_entries[ i ] is less than NumInpPicsInOutputTensor derived for the i-th NNPF of the target NNPFG, nnpfga\_output\_flag[ i ][ j ] is inferred to be equal to 1 for each value of i in the range of nnpfga\_num\_output\_entries[ i ] to NumInpPicsInOutputTensor − 1, inclusive.

Let NnpfgaOutputPicList, which is the list of pictures output by NNPF process of the NNPFG in output order, be initially empty and formed in decreasing order of n in the range of 0 to nnpfga\_num\_filters\_minus2 + 1, inclusive, by including each picture that is output by the NNPF process of the n-th loop entry that has no corresponding picture already present in NnpfgaOutputPicList.

**nnpfga\_num\_output\_pic\_update** specifies the number of output picture update flag that is present. When nnpfga\_num\_output\_pic\_update is greater than 0, nnpfga\_output\_pic\_update\_flag[ i ] for i in the range from 0 to nnpfga\_num\_output\_pic\_update − 1, inclusive, are present. The value of nnpfga\_num\_output\_pic\_update shall be in the range from 0 to number of pictures in NnpfgaOutputPicList, inclusive.

**nnpfga\_output\_pic\_update\_flag**[ i ] equal to 1 specifies that the i-th picture NnpfgaOutputPicList is included in the final output picture list. nnpfga\_output\_pic\_update\_flag[ i ] equal to 0 specifies that the i-th picture NnpfgaOutputPicList is not included in the final output picture list. When the value of nnpfga\_num\_output\_pic\_update is less than the number of pictures in NnpfgaOutputPicList, the value of nnpfga\_output\_pic\_update\_flag [ i ] for i in the range from nnpfga\_num\_output\_pic\_update to the number of pictures in NnpfgaOutputPicList − 1, inclusive, is inferred to be equal to 1.

Let numCandidateOutputPics be the number of pictures in NnpfgaOutputPicList. The list of output picture flag candOutputPicFlag[ i ] for i in the range from 0 to numCandidateOutputPics − 1, inclusive is derived as follows:

for( i = 0; i < numCandidateOutputPics; i++ )  
 candOutputPicFlag[ i ] = 1  
for( i = 0; i < nnpfga\_num\_output\_pic\_update; i++ ) (xx)  
 candOutputPicFlag[ i ] = nnpfga\_output\_pic\_update\_flag[ i ]

Let FinalNnpfgaOutputPicList, which is the list of pictures output by NNPF process of the NNPFG in output order, be derived as follows:

j = 0  
for( i = 0; i < numCandidateOutputPics; i++ )  
 if( candOutputPicFlag[ i ] ) (xx)  
 FinalNnpfgaOutputPicList[ j++ ] = NnpfgaOutputPicList[ i ]

*Alternative changes to VSEI subclause 8.28 and VVC subclause D.12.11 for adding the support of grouping of PFs (both NNPFs and non-NN PFs)*

Modify text of VSEI subclause 8.28 per JVET-AF0061 as follows (Additions highlighted in turquoise, and removals highlighted in ~~yellow strikethrough in red fonts~~):

**8.28 General post-processing filtering process and n~~N~~eural-network post-filter SEI messages**

**8.28.1 General post-processing filtering process ~~using NNPFs~~ using PPFs**

**8.28.1.1 General**

Input to this process is a bitstream BitstreamToFilter. Output of this process is ~~a list of NNPF output pictures ListNnpfOutputPics~~ a list of output pictures ListOutputPics.

A post-processing filter (PPF) may be indicated by an SEI message for which the payloadType value is in SeiProcessingOrderSeiList, specified in the semantics of the SEI processing order (SPO) SEI message.

For each picture, there can be multiple PPFs activated and belonging to one or more PPF groups. PPF groups are alternative to each other, i.e., at most one group can be chosen to be applied.

A special PPF cascading case is defined as the case when such two PPFs are both activated for a picture: the two PPFs are both NNPFs (i.e., the payloadType value for the PPFs indicates the NNPFC SEI message), one of the two NNPFs has nnpfc\_purpose equal to 4 and the other has multiple input pictures, and neither of the two NNPFs is associated with an SEI processing order SEI message. In this case, the two NNPFs are implicitly considered as belonging to one PPF group, and the NNPF with nnpfc\_purpose equal to 4 is applied first.

Except for the special PPF cascading case, each PPF group containing multiple PPFs is associated with an SPO SEI message with a particular value of po\_id.

Except for the special PPF cascading case, any PPF not associated with an SPO SEI message is in its own PPF group.

One or more PPFs in the chosen PPF group can be applied. When multiple PPFs (in the chosen PPF group) are applied, they are applied in the cascading manner, meaning that they are applied in the order indicated by the SEI processing order SEI message associated with the chosen PPF group, and for each applied PPF that is not the last applied PPF, the output is used as the input of the next applied PPF.

First, BitstreamToFilter is decoded, the list CroppedDecodedPictures is set to be the list of the cropped decoded pictures in output order resulted from decoding BitstreamToFilter, and the list ListOutputPics is initialized to be the same as CroppedDecodedPictures.

Second, the filtering process for one picture, as specified in subclause 8.28.1.2, is repeatedly invoked, in output order, for each cropped decoded picture that is in CroppedDecodedPictures and for which one or more ~~NNPFs~~ PPFs of one or more PPF groups are activated and only one of the groups is chosen to be applied.

The order of the pictures in ~~ListNnpfOutputPics~~ ListOutputPics is in output order.

Within ~~ListNnpfOutputPics~~ ListOutputPics there shall be no more than one picture pertaining to any particular output time instance. [Ed. Check phrasing of this. “Pertains” is not used in a similar way anywhere in the standard.] When for any particular picture in CroppedDecodedPictures there are ~~multiple NNPFs~~ PPFs of multiple PPF groups activated ~~and only one of the NNPFs is allowed to be chosen to be applied although any of the NNPFs may be chosen~~, the above constraint shall apply regardless of ~~which NNPF is chosen to be applied to the particular picture~~ which group of PPFs is chosen to be applied when the particular picture is the current picture.

For any particular pair of pictures inputPicA and inputPicB consecutive in output order in CroppedDecodedPictures, when there are one or more pictures interpolatedPicSetA in ~~ListNnpfOutputPics~~ ListOutputPics between inputPicA and inputPicB in output order, the pictures in interpolatedPicSetA shall be among the pictures that were output by applying a particular ~~NNPF nnpfA with PictureRateUpsamplingFlag equal to 1~~ PPF ppfA when a particular picture currPicA in CroppedDecodedPictures was the current picture. The application of any other ~~NNPF~~ PPF that was used in the filtering process for one picture when currPicA was the current picture or the application of any ~~NNPF (including nnpfA)~~ PPF (including ppfA) that was used in the filtering process for one picture when any other picture currPicB in CroppedDecodedPictures was the current picture shall not output any picture between the inputPicA and inputPicB in output order.

NOTE – The intent of the constraints expressed in the above paragraph is to disallow generating ~~NNPF~~ PPF output pictures between any particular pair of consecutive input pictures more than once.

**8.28.1.2 Filtering process for one picture**

The filtering process specified in this subclause applies to each cropped decoded picture, referred to as the current picture, that is in CroppedDecodedPictures and for which one or more groups of PPFs are activated, only one of the PPF groups is chosen to be applied, and the number of PPFs (in the chosen PPF group) to be applied is greater than 0.

The filtering process for one picture using one PPF, as specified in subclause 8.28.1.3, is repeatedly invoked for each of the PPFs to be applied. When the number of PPFs to be applied is greater than 1, the following applies:

– If the special PPF cascading case applies for the chosen PPF group, the NNPF with nnpfc\_purpose equal to 4 is applied first, followed by the NNPF with multiple input pictures.

– Otherwise (the special PPF cascading case does not apply for the chosen PPF group), the PPFs are applied in the preferred order indicated by the SEI processing order SEI message associated with the chosen group of PPFs.

**8.28.1.3 Filtering process for one picture using ~~an NNPF~~ one PPF**

The filtering process specified in this subclause applies ~~to each cropped decoded picture, referred to as the current picture, that is in CroppedDecodedPictures and for which one or more NNPFs are activated~~ when a particular PPF is applied when a particular picture is the current picture.

Before the PPF is applied, when the PPF is the first PPF to be applied, the list CandInputPicList is set to be identical to CroppedDecodedPictures.

When applying a PPF to the current picture, the input pictures for the PPF are selected from the list CandInputPicList, and the order of the pictures generated and output by the PPF are in output order.

When applying a PPF that is an NNPF to the current picture, the following applies:

– The filtered and/or interpolated pictures are generated by the NNPF by applying the NNPF process specified in the semantics of the NNPFC SEI message, in a patch-wise manner, to the current picture.

– The order of the pictures generated by the NNPF by applying the NNPF process being stored into the output tensor of the NNPF is in output order.

~~When the applied NNPF is the last NNPF that is applied to the current picture, the pictures generated by the NNPF and output by the NNPF process are included into ListNnpfOutputPics, in the same order as when the pictures are stored into the output tensor of the NNPF.~~

After the PPF is applied, the lists CandInputPicList and ListOutputPics are both updated, in the same manner, by 1) replacing each of those pictures in the list having a corresponding PPF output picture of the PPF with the corresponding PPF output picture, and 2) inserting those interpolated pictures, if any, into the list and placing them such that all pictures in the updated list are in output order.

**8.28.2 Neural-network post-filter characteristics SEI message**

**8.28.2.1 Neural-network post-filter characteristics SEI message syntax**

(no change)

**8.28.2.2 Neural-network post-filter characteristics SEI message semantics**

The neural-network post-filter characteristics (NNPFC) SEI message specifies a neural network that may be used as a post-processing filter. The use of specified neural-network post-processing filters (NNPFs) for specific pictures is indicated with neural-network post-filter activation (NNPFA) SEI messages.

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

– The list CandInputPicList that contains a list of pictures in output order from which the input pictures for the NNPF are selected.

NOTE 1 – This list is updated by the PPF filtering process each time when a PPF is applied for the current picture unless the PPF is the last PPF that is applied for the current picture.

– Input picture width and height in units of luma samples, denoted herein by CroppedWidth and CroppedHeight, respectively.

– Luma sample array CroppedYPic[ idx ] and chroma sample arrays CroppedCbPic[ idx ] and CroppedCrPic[ idx ], when present, of the input pictures with index idx in the range of 0 to numInputPics − 1, inclusive, that are used as input for the NNPF.

– Bit depth BitDepthY for the luma sample array of the input pictures.

– Bit depth BitDepthC for the chroma sample arrays, if any, of the input pictures.

~~– A chroma format indicator, denoted herein by ChromaFormatIdc, as described in subclause 7.3.~~

– Chroma format indicator ChromaFormatIdc, as described in subclause 7.3, of the input pictures.

– When nnpfc\_auxiliary\_inp\_idc is equal to 1, a filtering strength control value array StrengthControlVal[ idx ] that shall contain real numbers in the range of 0 to 1, inclusive, of the input pictures with index idx in the range of 0 to numInputPics − 1, inclusive.

Input picture with index 0 ~~corresponds~~ is the picture in CandInputPicList corresponding to the picture for which the NNPF defined by this NNPFC SEI message is activated by an NNPFA SEI message. Input picture with index i in the range of 1 to numInputPics − 1, inclusive, precedes the input picture with index i − 1 in output order.

NOTE 2 – The picture in CandInputPicList corresponding to a picture is either the cropped decoded output picture of that picture or a filtered version of the cropped decoded output picture that was an output picture of a previously applied PPF.

The variables SubWidthC and SubHeightC are derived from ChromaFormatIdc as specified by Table 2.

…

**nnpfc\_absent\_input\_pic\_zero\_flag** equal to 1 indicates that the NNPF expects an input picture corresponding to a picture that is not present in the bitstream to be represented by sample arrays with sample values equal to 0. nnpfc\_absent\_input\_pic\_zero\_flag equal to 0 indicates that the NNPF expects an input picture inputPicA corresponding to a picture that is not present in the bitstream to be represented by the input picture inputPicB that is the closest to inputPicA in output order and the picture corresponding to inputPicB is present in the bitstream.

…

**nnpfa\_no\_prev\_clvs\_flag** equal to 1 specifies that the pictures corresponding to the input pictures for the NNPF do not originate from a previous CLVS. nnpfa\_no\_prev\_clvs\_flag equal to 0 specifies that the pictures corresponding to the input pictures for the NNPF may or may not originate from a previous CLVS.

NOTE 4 – The value of nnpfa\_no\_prev\_clvs\_flag can be changed from 0 to 1, when the current CLVS is spliced from another bitstream next to the previous CLVS and this NNPFA SEI message would cause one or more input pictures to be selected that have corresponding pictures from one or more previous CLVSs and therefore is likely to impact the output of the target NNPF negatively.

**nnpfa\_no\_foll\_clvs\_flag** equal to 1 specifies that when this NNPFA SEI message persists for the last PU of a CLVS in output order, the NNPFA SEI message is treated like it persisted for the last PU, in output order, of the current layer within the bitstream. When this NNPFA SEI message does not persist for the last PU, in output order, of a CLVS in output order or nnpfa\_no\_foll\_clvs\_flag is equal to 0, the value of nnpfa\_no\_foll\_clvs\_flag causes no specific impact.

NOTE 5 – The value of nnpfa\_no\_foll\_clvs\_flag can be changed from 0 to 1 for a picture-rate-upsampling NNPF, when the following CLVS is spliced from a different bitstream next to the current CLVS. Consequently, the NNPF process interpolates pictures up to the end of the current CLVS using input pictures corresponding to pictures originating from the current CLVS only.

…

**8.28.3 Neural-network post-filter activation SEI message**

(no change)

Modify text of *VVC subclause D.12.11* per JVET-AF0061 as follows (Additions highlighted in turquoise, and removals highlighted in ~~yellow strikethrough in red fonts~~):

**D.12.11 Use of the post-processing filter SEI messages, including the neural network post-filter characteristics SEI message and the neural network post-filter activation SEI message**

A post-processing filter (PPF) may be indicated by an SEI message for which the payloadType value is in SeiProcessingOrderSeiList, which is specified in the semantics of the SEI processing order SEI message.

Let currPic be the cropped decoded output picture for which the post-processing filter (PPF), e.g., neural-network post-processing filter (NNPF) defined by the neural-network post-filter characteristics (NNPFC) SEI message, is activated, e.g., by a neural-network post-filter activation (NNPFA) SEI message, and currLayerId be the nuh\_layer\_id value of currPic.

The list candInputPicList contains a list of pictures in output order from which the input pictures for the PPF are selected.

NOTE 1 – This list is updated by the PPF filtering process each time when a PPF is applied for the current picture unless the PPF is the last PPF that is applied for the current picture.

When the PPF is not an NNPF, the PPF is considered to have only one input picture that is the picture in candInputPicList corresponding to the current picture.

NOTE 2 – The picture in candInputPicList corresponding to a cropped decoded output picture is either the cropped decoded output picture itself or a filtered version of the cropped decoded output picture that was an output picture of a previously applied PPF.

If the PPF is the first PPF that is applied for the current picture, the following applies:

– CroppedWidth is set equal to the value of pps\_pic\_width\_in\_luma\_samples − ‌SubWidthC \* ‌ ( pps\_conf\_win\_left\_offset + pps\_conf\_win\_right\_offset ) for currPic.

– CroppedHeight is set equal to the value of pps\_pic\_height\_in\_luma\_samples − ‌SubHeightC \* ‌ ( pps\_conf\_win\_top\_offset + pps\_conf\_win\_bottom\_offset ) for currPic.

– BitDepthY and BitDepthC are both set equal to BitDepth.

– ChromaFormatIdc is set equal to sps\_chroma\_format\_idc.

Otherwise (the PPF is not the first PPF that is applied for the current picture), the following applies:

– CroppedWidth is set equal to the picture width in units of luma samples of the picture in candInputPicList corresponding to the current picture.

– CroppedHeight is set equal to the picture height in units of luma samples of the picture in candInputPicList corresponding to the current picture.

– BitDepthY is set equal to the bit depth BitDepthY for the luma sample array of the picture in candInputPicList corresponding to the current picture.

– BitDepthC is set equal to the bit depth BitDepthC for the chroma sample arrays, if any, of the picture in candInputPicList corresponding to the current picture.

– ChromaFormatIdc is set equal to the chroma format indicator ChromaFormatIdc of the the picture in candInputPicList corresponding to the current picture.

The remainder of this subclause applies when the PPF is an NNPF.

It is a requirement of bitstream conformance that when a picture unit contains an NNPFA SEI message, the value of ph\_pic\_output\_flag in the picture header contained in that picture unit shall be equal to 1.

NOTE 3 – Since when the NNPF is the first PPF that is applied for currPic only cropped decoded output pictures are used as input pictures of the NNPF, the value of ph\_pic\_output\_flag in the picture header of the coded picture corresponding to each input picture of the NNPF is equal to 1.

The variable pictureRateUpsamplingFlag is set equal to ( ( nnpfc\_purpose & 0x08 ) > 0 ) ? 1 : 0.

The variable numInputPics is set equal to nnpfc\_num\_input\_pics\_minus1 + 1.

The variable numInferences is derived as follows:

– If all of the following conditions are true, the variable numPostRoll is set equal to the value of i such that nnpfc\_interpolated\_pics[ i ] is greater than 0 and the variable numInferences is set equal to 1 + numPostRoll:

– nnpfc\_purpose is equal to 8 (i.e., the only purpose for the NNPF is picture rate upsampling).

– nnpfa\_persistence\_flag is equal to 1.

– nnpfc\_interpolated\_pics[ i ] is greater than 0 only for a single value of i that is greater than 0.

– Either of the following conditions is true:

– currPic is the last picture of the bitstream in output order that has nuh\_layer\_id equal to currLayerId.

– currPic is the last picture in the CLVS in output order and nnpfa\_no\_foll\_clvs\_flag is equal to 1.

– Otherwise, if all of the following conditions are true, the variable numPostRoll is set equal to InpIdx[ i ] for the value of i such that nnpfa\_output\_flag[ i ] is equal to 1, and the variable numInferences is set equal to 1 + numPostRoll:

– pictureRateUpsamplingFlag is equal to 0.

– numInputPics is greater than 1.

– nnpfa\_persistence\_flag is equal to 1.

– nnpfa\_output\_flag[ idx ] is equal to 1 for a single value of idx in the range of 0 to NumInpPicsInOutputTensor − 1, inclusive, and for that single value of idx, InpIdx[ idx ] is greater than 0.

– Either of the following conditions is true:

– currPic is the last picture of the bitstream in output order that has nuh\_layer\_id equal to currLayerId.

– currPic is the last picture in the CLVS in output order and nnpfa\_no\_foll\_clvs\_flag is equal to 1.

– Otherwise, the variable numInferences is set equal to 1.

For each value of j in the range of 0 to numInferences − 1, inclusive, the following applies for the derivation of the input pictures for the NNPF such that each input picture is a picture in candInputPicList:

– The arrays inputPic[ i ] and inputPresentFlag[ i ] for i in the range of 0 to numInputPics − 1, inclusive, representing all the input pictures and the presence of input pictures, respectively, are specified as follows:

– When j is greater than 0, for each value of k in the range of 0 to j − 1, inclusive, inputPic[ k ] is set to be the picture in candInputPicList corresponding to currPic and inputPresentFlag[ k ] is set equal to 0.

– The j-th input picture, inputPic[ j ], is set to be the picture in candInputPicList corresponding to currPic and inputPresentFlag[ j ] is set equal to 1.

– When numInputPics is greater than 1, the following applies for each value of i in the range of j + 1 to numInputPics − 1, inclusive, in increasing order of i:

– If both of the following conditions are true, inputPic[ i ] is set to be the picture in candInputPicList corresponding to prevPic and inputPresentFlag[ i ] is set equal to 1:

– Either of the following conditions is true:

– pictureRateUpsamplingFlag is equal to 1 and currPic is associated with a frame packing arrangement SEI message with frame\_packing\_arrangement\_type equal to 5 and a particular value of fp\_current\_frame\_is\_frame0\_flag, and there is a cropped decoded output picture prevPic that is the last picture in output order among all cropped decoded output pictures that have nuh\_layer\_id equal to currLayerId, precede inputPic[ i − 1 ] in output order, and are associated with a frame packing arrangement SEI message with frame\_packing\_arrangement\_type equal to 5 and the same value of fp\_current\_frame\_is\_frame0\_flag.

– pictureRateUpsamplingFlag is equal to 0 or currPic is not associated with a frame packing arrangement SEI message with frame\_packing\_arrangement\_type equal to 5, and there is a cropped decoded output picture prevPic that is the last picture in output order among all cropped decoded output pictures that have nuh\_layer\_id equal to currLayerId and precede inputPic[ i − 1 ] in output order.

– nnpfa\_no\_prev\_clvs\_flag is equal to 0 or the coded picture corresponding to prevPic and the current picture are present in the same CLVS.

– Otherwise, the following applies:

– inputPic[ i ] is set to be the same picture as inputPic[ i − 1 ] and inputPresentFlag[ i ] is set equal to 0.

– It is a requirement of bitstream conformance that, when pictureRateUpsamplingFlag is equal to 1, nnpfc\_interpolated\_pics[ i − 1 ] shall be equal to 0.

– It is a requirement of bitstream conformance that when inputPresentFlag[ i ] is equal to 0 and nnpfc\_input\_pic\_output\_flag[ i ] is equal to 1, the value of nnpfa\_output\_flag[ idx ] shall be equal to 0 for the value of idx such that InpIdx[ idx ] is equal to i.

– For purposes of interpretation of the NNPFC SEI message, the following variables are specified:

~~– If numInputPics is greater than 1 and there is a second NNPF that is defined by at least one NNPFC SEI message, is activated by an NNPFA SEI message for currPic, and has nnpfc\_purpose equal to 4, the following applies:~~

~~– CroppedWidth is set equal to nnpfcOutputPicWidth defined for the second NNPF.~~

~~– CroppedHeight is set equal to nnpfcOutputPicHeight defined for the second NNPF.~~

~~– Otherwise, the following applies:~~

~~– CroppedWidth is set equal to the value of pps\_pic\_width\_in\_luma\_samples − ‌SubWidthC \* ‌ ( pps\_conf\_win\_left\_offset + pps\_conf\_win\_right\_offset ) for currPic.~~

~~– CroppedHeight is set equal to the value of pps\_pic\_height\_in\_luma\_samples − ‌SubHeightC \* ‌ ( pps\_conf\_win\_top\_offset + pps\_conf\_win\_bottom\_offset ) for currPic.~~

– The luma sample arrays CroppedYPic[ i ] and the chroma sample arrays CroppedCbPic[ i ] and CroppedCrPic[ i ], when present, are derived as follows for each value of i in the range of 0 to numInputPics − 1, inclusive:

– The variable sourcePic is derived as follows:

– If inputPresentFlag[ i ] is equal to 1 or nnpfc\_absent\_input\_pic\_zero\_flag is equal to 0, sourcePic is set to be inputPic[ i ].

– Otherwise (inputPresentFlag[ i ] is equal to 0 and nnpfc\_absent\_input\_pic\_zero\_flag is equal to 1), sourcePic is set to be a picture with a luma sample array of CroppedWidth × CroppedHeight samples equal to 0 and Cb and Cr sample arrays of ( CroppedWidth / SubWidthC ) × ( CroppedHeight / SubHeightC ) samples equal to 0.

~~– If numInputPics is equal to 1, the following applies:~~

– The luma sample array CroppedYPic[ i ] and the chroma sample arrays CroppedCbPic[ i ] and CroppedCrPic[ i ], when present, are set to be the 2-dimensional arrays of decoded sample values of the Y, Cb and Cr components, respectively, of sourcePic.

~~– Otherwise (numInputPics is greater than 1), the following applies:~~

~~– The variable sourceWidth is set equal to the value of pps\_pic\_width\_in\_luma\_samples − ‌SubWidthC \* ( pps\_conf\_win\_left\_offset + pps\_conf\_win\_right\_offset ) for sourcePic.~~

~~– The variable sourceHeight is set equal to the value of pps\_pic\_height\_in\_luma\_samples − ‌SubHeightC \* ( pps\_conf\_win\_top\_offset + pps\_conf\_win\_bottom\_offset ) for sourcePic.~~

~~– If sourceWidth is equal to CroppedWidth and sourceHeight is equal to CroppedHeight, resampledPic is set to be the same as sourcePic.~~

~~– Otherwise (sourceWidth is not equal to CroppedWidth or sourceHeight is not equal to CroppedHeight), the following applies:~~

~~– There shall be an NNPF, hereafter referred to as the super resolution NNPF, that is defined by at least one NNPFC SEI message, is activated by an NNPFA SEI message for sourcePic, and has nnpfc\_purpose equal to 4, nnpfcOutputPicWidth equal to CroppedWidth and nnpfcOutputPicHeight equal to CroppedHeight.~~

~~– resampledPic is set to be the output of the neural-network inference of the super resolution NNPF with sourcePic being an input.~~

~~– The luma sample array CroppedYPic[ i ] and the chroma sample arrays CroppedCbPic[ i ] and CroppedCrPic[ i ], when present, are set to be the 2-dimensional arrays of decoded sample values of the Y, Cb and Cr components, respectively, of resampledPic.~~

~~– BitDepth~~~~Y~~ ~~and BitDepth~~~~C~~ ~~are both set equal to BitDepth.~~

~~– ChromaFormatIdc is set equal to sps\_chroma\_format\_idc.~~

– The array StrengthControlVal[ i ] for all values of i in the range of 0 to numInputPics − 1, inclusive, specifying the filtering strength control value for the input pictures for the NNPF, is derived as follows:

– StrengthControlVal[ i ] is set equal to the value of ( firstSliceQpY + QpBdOffset ) ÷ ( 63 + QpBdOffset ), where firstSliceQpY is equal to SliceQpY of the first slice of the cropped decoded output picture corresponding to inputPic[ i ].

There shall not be more than two NNPFC SEI messages present in a picture unit with the same value of nnpfc\_id. When there are two NNPFC SEI messages present in a picture unit with the same value of nnpfc\_id, these SEI messages shall have different content. When two NNPFC SEI messages with the same nnpfc\_id and different content are present in the same picture unit, both of these NNPFC SEI messages shall be in the same SEI NAL unit.

*Addition of post-processing filter gain SEI message*

* + 1. **Post-processing filter gain SEI message**

[Ed. (YK): Note that the JVET agreement was "to include the approach of separate SEI (applicable for any PF) to TuC". Therefore, this SEI message should be renamed and changed accordingly such that it applies to both NN post-processing filters and non-NN post-processing filters. And the subclause should not be a subclause of 8.28. For example, it could be subclause 8.35. (MH): My mistake, sorry for the oversight, and thanks for catching the deviation from the documented adoption. I have edited the text to make the SEI message generic to any type of post-filters. I didn't move the text yet to out from 8.28 to make it easier to review my edits to generalize this SEI message to apply to any type of post-filters. Also, we may consider generalizing 8.28 to be an umbrella for all post-filter related SEI messages and subclauses, now that 8.28.1 is also generic to any type of post-filters.]

* + - 1. **Post-propcessing filter gain SEI message syntax**

|  |  |
| --- | --- |
| post\_processing\_filter\_gain( payloadSize ) { | **Descriptor** |
| **pf\_gain\_id** | ue(v) |
| **pf\_gain\_num\_entries\_minus1** | ue(v) |
| for( i = 0; i <= pf\_gain\_num\_entries\_minus1; i++ ) |  |
| **pf\_gain\_metric\_idc**[ i ] | ue(v) |
| for( i = 0; i <= pf\_gain\_num\_entries\_minus1; i++ ) { |  |
| **pf\_gain\_per\_component\_flag**[ i ] | u(1) |
| if( pf\_gain\_per\_component\_flag[ i ] ) { |  |
| **pf\_gain\_y\_flag**[ i ] | u(1) |
| **pf\_gain\_cb\_flag**[ i ] | u(1) |
| if( pf\_gain\_y\_flag[ i ] + pf\_gain\_cb\_flag[ i ] > 0 ) |  |
| **pf\_gain\_cr\_flag**[ i ] | u(1) |
| } |  |
| **pf\_gain\_word\_cnt\_minus1**[ i ] | ue(v) |
| for( j = 0; j < pfGainNumValuesPerEntry[ i ]; j++ ) |  |
| **pf\_gain\_value**[ i ][ j ] | u(v) |
| } |  |
| } |  |

* + - 1. **Post-processing filter gain SEI message semantics**

The post-processing filter gain SEI message indicates the quality improvement that is achieved by applying an indicated post-processing filter (PPF) or an indicated PPF group.

**pf\_gain\_id** indicates the PPF or PPF group which this SEI message is associated with. When pf\_gain\_id is equal to any nnpfc\_id or nnpfgc\_id value present in the current CLVS, this SEI message indicates one or more values characterizing the gain obtained by the NNPF identified by the nnpfc\_id value or the NNPFG identified by the nnpfgc\_id value, respectively. The variables specified for the NNPF with nnpfc\_id equal to nnpf\_gain\_id or the NNPFG with nnpfgc\_id equal to nnpf\_gain\_id apply below.

**pf\_gain\_num\_entries\_minus1** plus 1 specifies the number of entries in this syntax structure.

**pf\_gain\_metric\_idc**[ i ] identifies the metric used in the gain values provided for the i-th entry in this syntax structure. pf\_gain\_metric\_idc[ i ] equal to 0 indicates a peak signal-to-noise ratio based metric specified below. The value of pf\_gain\_metric\_idc[ i ] shall be in the range of 0 to 65 535, inclusive. The value of pf\_gain\_metric\_idc[ i ] shall be equal to 0 in bitstreams conforming to this version of this Specification. Values of 1 to 65 535, inclusive, for pf\_gain\_metric\_idc[ i ] are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this Specification. When the value of pf\_gain\_metric\_idc[ i ] is in the range of 1 to 65 535, inclusive, decoders conforming to this version of this Specification shall ignore all the syntax elements for the i-th entry in this syntax structure.

**pf\_gain\_per\_component\_flag**[ i ] equal to 1 specifies that gain values are provided on colour component basis. pf\_gain\_per\_component\_flag[ i ] equal to 0 specifies that the i-th entry in this syntax structure has one and only one gain value, which may be derived collectively from several colour components.

**pf\_gain\_y\_flag**[ i ], when present and equal to 1, specifies that the 0-th gain value for the i-th entry in this syntax structure is derived from the luma component. pf\_gain\_y\_flag[ i ], when present and equal to 0, specifies that the i-th entry in this syntax structure does not have a gain value derived from the luma component.

**pf\_gain\_cb\_flag**[ i ], when present and equal to 1, specifies that the ( pf\_gain\_y\_flag[ i ] )-th gain value for the i-th entry in this syntax structure is derived from the Cb component. pf\_gain\_cb\_flag[ i ], when present and equal to 0, specifies that the i-th entry in this syntax structure does not have a gain value derived from the Cb component.

**pf\_gain\_cr\_flag**[ i ], when present and equal to 1 or when inferred to be equal to 1, specifies that the ( pf\_gain\_y\_flag[ i ] + pf\_gain\_cb\_flag[ i ] )-th gain value for the i-th entry in this syntax structure is derived from the Cr component. pf\_gain\_cr\_flag[ i ], when present and equal to 0, specifies that the i-th entry in this syntax structure does not have a gain value derived from the Cr component. When pf\_gain\_y\_flag[ i ] and pf\_gain\_cb\_flag[ i ] are present and both equal to 0, pf\_gain\_cr\_flag[ i ] is inferred to be equal to 1.

The value of pfGainNumValuesPerEntry[ i ], which specifies the count of gain values for the i-th entry in this syntax structure, is derived as follows:

pfGainNumValuesPerEntry[ i ] = pf\_gain\_per\_component\_flag[ i ] ?  
 pf\_gain\_y\_flag[ i ] + pf\_gain\_cb\_flag[ i ] + pf\_gain\_cr\_flag[ i ] : 1 (xx)

**pf\_gain\_word\_cnt\_minus1**[ i ] plus 1 specifies the length of each gain value for the i-th entry in units of 16 bits.

**pf\_gain\_value**[ i ][ j ] specifies the j-th gain value for the i-th entry in this syntax structure. The semantics and the representation format of the gain value are determined by the value of pf\_gain\_metric\_idc[ i ]. The length of pf\_gain\_value[ i ][ j ] is 16 \* ( pf\_gain\_word\_cnt\_minus1[ i ] + 1 ) bits.

When pf\_gain\_metric\_idc[ i ] is equal to 0, it is a requirement of bitstream conformance that all the following constraints apply:

– pf\_gain\_per\_component\_flag[ i ] is equal to 1.

– pf\_gain\_word\_cnt\_minus1[ i ] is equal to 0.

– nnpfc\_out\_format\_idc is equal to 1.

– When pf\_gain\_y\_flag[ i ] is equal to 1, all of the following constraints apply:

– nnpfcOutputPicWidth is equal to CroppedWidth.

– nnpfcOutputPicHeight is equal to CroppedHeight.

– When pf\_gain\_cb\_flag[ i ] is equal to 1 or pf\_gain\_cr\_flag[ i ] is equal to 1, all of the following constraints apply:

– nnpfcOutputPicWidth / outSubWidthC is equal to CroppedWidth / SubWidthC.

– nnpfcOutputPicHeight / outSubHeightC is equal to CroppedHeight / SubHeightC.

The function Log10( x ) is defined as the base-10 logarithm of x.

Given sample arrays origSampleArray and testSampleArray, the function Psnr( origSampleArray, testSampleArray, width, height, origBitDepth, testBitDepth ) returns the value of psnrVal, which is the average peak signal-to-noise ratio of testSampleArray in comparison to origSampleArray, derived as follows:

maxBitDepth = (origBitDepth > testBitDepth) ? origBitDepth : testBitDepth  
origShift = maxBitDepth − origBitDepth  
testShift = maxBitDepth − testBitDepth  
maxVal = 255 << ( maxBitDepth − 8 )  
maxSquaredVal = maxVal \* maxVal  
sumSquaredError = 0  
for( y = 0; y < height; y++ )  
 for( x = 0; x < width; x++ ) {  
 sampleError =   
 ( testSampleArray[ x ][ y ] << testShift ) − ( origSampleArray[ x ][ y ] << origShift ) (xx)  
 sumSquaredError += sampleError \* sampleError  
 }  
if( sumSquaredError = = 0 )  
 psnrVal = 999.99  
else  
 psnrVal = 10 \* Log10( maxSquaredVal ÷ ( sumSquaredError ÷ width ÷ height ) )

When pf\_gain\_metric\_idc[ i ] is equal to 0, the following applies

– Let OriginalPictures be the list of pictures given as input to encoding in output order and OrigBitDepth be the bit depth of the sample arrays of the pictures in OriginalPictures.

– CroppedDecodedPictures and ListOutputPics are derived as specified in subclause 8.28.1.1.

– It is a requirement of bitstream conformance that the number of pictures, assigned to variable numPics, in OriginalPictures, CroppedDecodedPictures and ListOutputPics is the same.

– avgPsnrDiff is derived as follows:

sumPsnrDiff = 0  
for( a = 0; a < numPics; a++ ) {  
 if( j = = 0 && pf\_gain\_y\_flag [ i ] = = 1 )  
 cIdx = 0  
 else if( ( j = = 0 && pf\_gain\_y\_flag[ i ] = = 0 && pf\_gain\_cb\_flag[ i ] = = 1 ) | |   
 ( j = = 1 && pf\_gain\_y\_flag[ i ] = = 1 && pf\_gain\_cb\_flag[ i ] = = 1 )  
 cIdx = 1  
 else  
 cIdx = 2  
 if( cIdx = = 0 ) { (xx)  
 arrayWidth = CroppedWidth  
 arrayHeight = CroppedHeight  
 } else {  
 arrayWidth = CroppedWidth / SubWidthC  
 arrayHeight = CroppedHeight / SubHeightC  
 }  
 if( cIdx = = 0 )  
 sumPsnrDiff += Psnr( OriginalPictures[ a ][ cIdx ], ListOutputPics[ a ][ cIdx ], width, height,   
 OrigBitDepth, outTensorBitDepthY ) − Psnr( OriginalPictures[ a ][ cIdx ],   
 CroppedDecodedPictures[ a ][ cIdx ], width, height, OrigBitDepth, BitDepthY )  
 else  
 sumPsnrDiff += Psnr( OriginalPictures[ a ][ cIdx ], ListOutputPics[ a ][ cIdx ], width, height,   
 OrigBitDepth, outTensorBitDepthC ) − Psnr( OriginalPictures[ a ][ cIdx ],   
 CroppedDecodedPictures[ a ][ cIdx ], width, height, OrigBitDepth, BitDepthC )  
}  
avgPsnrDiff = sumPsnrDiff ÷ numPics

– If avgPsnrDiff is greater than or equal to 256, pf\_gain\_value[ i ][ j ] is set equal to 65 535. Otherwise (avgPsnrDiff is less than 256), pf\_gain\_value[ i ][ j ] is set to a value such that pf\_gain\_value[ i ][ j ] >> 8 specifies the integer part of avgPsnrDiff and pf\_gain\_value[ i ][ j ] & 255 specifies the fractional part of avgPsnrDiff.

*Text to be added to a future edition or ammendment of Recommendation ITU-T H.266 | International Standard ISO/IEC 23090-3:*

**Use of the post-processing filter gain SEI message in VVC**

When there is a post-processing filter gain SEI message that has pf\_gain\_id equal to any nnpfc\_id or nnpfgc\_id value in the same CLVS, it shall be present in an SEI NAL unit that contains an NNPFC SEI message with nnpfc\_id equal to pf\_gain\_id or an NNPFGC SEI message with nnpfgc\_id equal to to pf\_gain\_id.

*Text to be added to a future edition or ammendment of Recommendation ITU-T H.265 | International Standard ISO/IEC 23008-2:*

**Use of the post-processing filter gain SEI message in HEVC**

When there is a post-processing filter gain SEI message that has pf\_gain\_id equal to any nnpfc\_id or nnpfgc\_id value in the same CLVS, it shall be present in an SEI NAL unit that contains an NNPFC SEI message with nnpfc\_id equal to pf\_gain\_id or an NNPFGC SEI message with nnpfgc\_id equal to pf\_gain\_id.

*Text to be added to a future edition or ammendment of Recommendation ITU-T H.264 | International Standard ISO/IEC 14496-10:*

**Use of the post-processing filter gain SEI message in AVC**

When there is a post-processing filter gain SEI message that has pf\_gain\_id equal to any nnpfc\_id or nnpfgc\_id value in the same CVS, it shall be present in an SEI NAL unit that contains an NNPFC SEI message with nnpfc\_id equal to pf\_gain\_id or an NNPFGC SEI message with nnpfgc\_id equal to pf\_gain\_id.

Renumber the current subclause 8.30 as 8.35 and add subclauses 8.30 to 8.34 as follows:

*Add VSEI subclause 8.30*

Modify text of JVET-AE2032 subclause 8.30 per JVET-AF0052 and JVET-AF0107 as follows

* 1. **Encoder optimization information SEI message**
     1. **Encoder optimization information SEI message syntax**

|  |  |
| --- | --- |
| encoder\_optimization\_info(payloadSize ) { | **Descriptor** |
| **eoi\_cancel\_flag** | u(1) |
| if( !eoi\_cancel\_flag ) { |  |
| **eoi\_persistence\_flag** | u(1) |
| **eoi\_for\_human\_viewing\_flag** | u(1) |
| **eoi\_for\_machine\_analysis\_flag** | u(1) |
| **eoi\_type** | u(16) |
| if( EoiObjectBasedFlag ) |  |
| **eoi\_object\_based\_idc** | ue(v) |
| if( EoiTemporalResamplingFlag ) { |  |
| **eoi\_temporal\_resampling\_type\_flag** | u(1) |
| **eoi\_num\_int\_pics** | ue(v) |
| } |  |
| } |  |
| } |  |

* + 1. **Encoder optimization information SEI message semantics**

The encoder optimization information SEI message is used to indicate if the video has been optimized for human viewing or machine analysis and which types of optimization have been applied in pre-processing or encoding.

**eoi\_cancel\_flag** equal to 1 specifies that the persistence of the encoder optimization information SEI message included in any previous PU in output order is cancelled. eoi\_cancel\_flag equal to 0 indicates that information on optimization that has been applied in pre-processing or encoding follows.

**eoi\_persistence\_flag** specifies the persistence of the optimization information provided in this SEI message. eoi\_persistence\_flag equal to 0 specifies that the optimization information applies for the current picture only. eoi\_persistence\_flag equal to 1 specifies that the optimization information applies for the current picture and 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 associated with an encoder optimization information SEI message is output that follows the current picture in output order.

**eoi\_for\_human\_viewing\_flag** equal to 1 specifies that purposes for the applied optimization include human viewing. eoi\_for\_human\_viewing\_flag equal to 0 specifies that purposes for the applied optimization may or may not include human viewing.

**eoi\_for\_machine\_analysis\_flag** equal to 1 specifies that purposes for the applied optimization include machine analsysis. eoi\_for\_machine\_analysis\_flag equal to 0 specifies that purposes for the applied optimization may or may not include machine analysis.

**eoi\_type** indicates the types of optimization method as specified in Table x1 where ( eoi\_type & bitMask ) not equal to 0 indicates that the optimization type with the bitMask value in Table x1 has been applied. When eoi\_type is greater than 0 and ( eoi\_type & bitMask ) is equal to 0, the optimization type with the bitMask value has not been applied. When eoi\_type is equal to 0, optimization as determined by the application has been used.

**Table x1 – Definition of** **eoi\_type**

|  |  |
| --- | --- |
| **bitMask** | **Interpretation** |
| 0x01 | Object-based optimization; the pictures for which this SEI message persists have been pre-processed or encoded so that detected objects in the pictures are optimized with respect to other parts of the pictures for the indicated optimization purposes |
| 0x02 | Temporal resampling optimization |
| 0x04 | Spatial resampling optimization |
| 0x08 | Temporal quality optimization in a manner that quality fluctuates temporally |
| 0x10 | Spatial quality optimization; the pictures for which this SEI message persists have been pre-processed or encoded to reduce unnecessary information or improve the quality of necessary information.(e.g to reduce the amount of noise and remove speckles at the picture-level) |

The variables EoiObjectBasedFlag, EoiTemporalResamplingFlag, EoiSpatialResamplingFlag, EoiTemporalQualityFlag, and EoiSpatialQualityFlag, specifying whether eoi\_type indicates the type of the optimization to include object-based optimization, temporal resampling optimization, spatial resampling optimization, temporal quality optimization and spatial quality optimization, respectively, are derived as follows:

EoiObjectBasedFlag = ( ( eoi\_type & 0x01 ) > 0 ) ? 1 : 0  
EoiTemporalResamplingFlag = ( ( eoi\_type & 0x02 ) > 0 ) ? 1 : 0  
EoiSpatialResamplingFlag = ( ( eoi\_type & 0x04 ) > 0 ) ? 1 : 0 (xx)  
EoiTemporalQualityFlag = ( ( eoi\_type & 0x08 ) > 0 ) ? 1 : 0  
EoiSpatialQualityFlag = ( ( eoi\_type & 0x10 ) > 0 ) ? 1 : 0

NOTE – For example, when certain highest temporal sublayers have been encoded with such coarse quantization that human viewers perceive the quality fluctuation annoying, but machine task performance is not compromised, eoi\_for\_human\_viewing\_flag and eoi\_for\_machine\_analaysis\_flag can be set equal to 0 and 1, respectively, and eoi\_type can be set equal to a value that causes EoiTemporalQualityFlag to be equal to 1.

When eoi\_persistence\_flag is equal to 0, it is a requirement of bitstream conformance that EoiTemporalResamplingFlag shall be equal to 0 and EoiTemporalQualityFlag shall be equal to 0.

**eoi\_object\_based\_idc**, when present, indicates the type of object-based optimization as specified in Table x2, where ( eoi\_object\_based\_idc & bitMask ) not equal to 0 indicates that the object-based optimization type associated with the bitMask value in Table x2 has been applied. When eoi\_object\_based\_idc is greater than 0 and ( eoi\_object\_based\_idc & bitMask ) is equal to 0, the object-based optimization type associated with the bitMask value has not been applied. When eoi\_object\_based\_idc is equal to 0, an application-defined type of object-based optimization has been applied. The value of eoi\_object\_based\_idc shall be in the range of 0 to 7, inclusive, in bitstreams conforming to this version of this Specification. Values of 8 to 65 535, inclusive, for eoi\_object\_based\_idc are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this Specification. When the value of eoi\_object\_based\_idc is in the range of 8 to 65 535, inclusive, decoders conforming to this version of this Specification shall ignore eoi\_object\_based\_idc.

**Table x2 – Definition of** **eoi\_object\_based\_idc**

|  |  |
| --- | --- |
| **bitMask** | **Interpretation** |
| 0x01 | Areas outside the detected objects have been blurred prior to encoding. |
| 0x02 | Areas outside the detected objects have been encoded with coarser transform-domain quantization than the quantization used for the detected objects. |
| 0x04 | Areas outside the detected objects have been overwritten. For example, an encoding system can overwrite areas outside the detected objects with a constant sample value. |

**eoi\_temporal\_resampling\_type\_flag** equal to 0 specifies that the temporal resampling optimization is a subsampling operation. eoi\_temporal\_resampling\_type\_flag equal to 1 specifies that the temporal resampling optimization is an upsampling operation.

**eoi\_num\_int\_pics** greater than 0 indicates that the count of pictures that the encoding system excluded between each pair of coded pictures in output order (when eoi\_temporal\_resampling\_type\_flag is equal to 0) or added between each pair of source pictures for encoding (when eoi\_temporal\_resampling\_type\_flag is equal to 1) within the persistence of this SEI message is constant. When eoi\_temporal\_resampling\_type\_flag is equal to 0 and eoi\_num\_int\_pics is greater than 0, eoi\_num\_int\_pics specifies the count of pictures that the encoding system excluded between each pair of coded pictures in output order. When eoi\_temporal\_resampling\_type\_flag is equal to 1 and eoi\_num\_int\_pics is greater than 0, eoi\_num\_int\_pics specifies the count of pictures that the encoding system added between each pair of source pictures for encoding.

eoi\_num\_int\_pics equal to 0 indicates that the count of pictures that the encoding system excluded between each pair of coded pictures in output order (when eoi\_temporal\_resampling\_type\_flag is equal to 0) or added between each pair of source pictures for encoding (when eoi\_temporal\_resampling\_type\_flag is equal to 1) within the persistence of this SEI message is unknown or varying.

The value of eoi\_num\_int\_pics shall be in the range of 0 to 63, inclusive.

*Add VSEI subclause 8.31*

Modify text of JVET-AE2032 subclause 8.31 per JVET-AF0055, JVET-AF0069 and JVET-AF0097 as follows

* 1. **Source picture timing information SEI message**
     1. **Source picture timing information SEI message syntax**

|  |  |
| --- | --- |
| source\_picture\_timing\_info( payloadSize ) { | **Descriptor** |
| **spti\_cancel\_flag** | u(1) |
| if( !spti\_cancel\_flag ) { |  |
| **spti\_persistence\_flag** | u(1) |
| **spti\_source\_timing\_equals\_output\_timing\_flag** | u(1) |
| if( !spti\_source\_timing\_equals\_output\_timing\_flag ) { |  |
| **spti\_source\_type\_present\_flag** | u(1) |
| if( spti\_source\_type\_present\_flag ) |  |
| **spti\_source\_type** | u(16) |
| **spti\_time\_scale** | u(32) |
| **spti\_num\_units\_in\_elemental\_interval** | u(18) |
| if( spti\_persistence\_flag ) |  |
| **spti\_max\_sublayers\_minus\_1** | u(3) |
| for( i = 0; i  <=  spti\_max\_sublayers\_minus1; i++ ) { |  |
| **spti\_sublayer\_interval\_scale\_factor**[ i ] | ue(v) |
| **spti\_sublayer\_synthesized\_picture\_flag**[ i ] | u(1) |
| } |  |
| } |  |
| **}** |  |
| } |  |

* + 1. **Source picture timing information SEI message semantics**

The source picture timing information (SPTI) SEI message indicates the temporal distance between source pictures associated with the corresponding decoded output pictures prior to encoding, e.g., for camera-captured content, the temporal distance between source pictures is the difference between the time at which an image sensor was exposed to produce a source picture associated with the current decoded picture and the time at which the image sensor was exposed to produce the source picture associated with a previous decoded picture in output order. The information provided by the SPTI SEI message pertains only for picture(s) starting from the picture in the current layer in the access unit that contains the SPTI SEI message and all subsequent pictures of the current layer in output order based on its persistence. [Ed. Check phrasing of this. “Pertains” is not used in a similar way anywhere in the standard.]

**spti\_cancel\_flag** equal to 1 indicates that the SPTI SEI message cancels the persistence of any previous SPTI SEI message in output order that applies to the current layer. spti\_cancel\_flag equal to 0 indicates that source picture timing information follows.

**spti\_persistence\_flag** specifies the persistence of the SPTI SEI message for the current layer.

spti\_persistence\_flag equal to 0 specifies that the SPTI SEI message applies to the current decoded picture only.

spti\_persistence\_flag equal to 1 specifies that the SPTI 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 an SPTI SEI message is output that follows the current picture in output order.

**spti\_source\_timing\_equals\_output\_timing\_flag** equal to 1 indicates the timing of source pictures is the same as the timing of corresponding decoded output pictures. spti\_source\_timing\_equals\_output\_timing\_flag equal to 0 indicates the timing of source pictures might not be the same as the timing of corresponding decoded output pictures.

When spti\_source\_timing\_equals\_output\_timing\_flag is equal to 1 and a picture timing SEI message is present for the current picture, source picture timing could be determined from information conveyed in the picture timing SEI message.

**spti\_source\_type\_present\_flag** equal to 1 indicates the syntax element spti\_source\_type is present in the SEI message. spti\_source\_type\_present\_flag equal to 0 indicates the syntax element spti\_source\_type is not present in the SEI message.

**spti\_source\_type** indicates the timing relationship between source pictures and corresponding decoded output pictures as specified in Table X, where ( spti\_source\_type & bitMask ) not equal to 0 indicates that the timing relationship has the interpretation associated with the bitMask value in Table X. When spti\_source\_type is greater than 0 and ( spti\_source\_type & bitMask ) is equal to 0, the interpretation associated with the bitMask value is not applicable to the SPTI SEI message. When spti\_source\_type is equal to 0, the timing relationship may be specified by the application.

The value of spti\_source\_type shall be in the range of 0 to 127, inclusive, in bitstreams conforming to this edition of this document. Values of 128 to 255, inclusive, for spti\_source\_type are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this edition of this document. Decoders conforming to this edition of this document shall ignore SPTI SEI messages with spti\_source\_type in the range of 128 to 255, inclusive.

**Table X – Interpretation of spti\_source\_type**

|  |  |
| --- | --- |
| **bitMask** | **Interpretation** |
| 0x01 | Slow motion: The absolute value of the temporal distance between consecutive source pictures is likely to be less than the temporal distance between corresponding decoded output pictures. |
| 0x02 | Sped-up motion: The absolute value of the temporal distance between consecutive source pictures is likely to be greater than the temporal distance between corresponding decoded output pictures. |
| 0x04 | High-speed imaging: The absolute value of the temporal distance between consecutive source pictures is likely to be less than 1/120 seconds. |
| 0x08 | Time-lapse imaging: The temporal distance between source pictures is likely to be greater than 1.001/24 seconds. |
| 0x10 | Temporal reversal: The absolute value of the temporal distance between consecutive source pictures is indicated to be negative (i.e., decoded pictures are output in reverse temporal order relative to the timing of the corresponding source pictures). |
| 0x20 | Still image / freeze frame: The temporal distance between source pictures is likely to be 0 (i.e., two or more decoded pictures are likely to represent the same source picture). |
| 0x40 | Sporadic or event-driven: The temporal distance between source pictures is likely to be non-constant. |

The value of ( spti\_source\_type & 0x04 ) & ( spti\_source\_type & 0x08 ) shall be zero (i.e., spti\_source\_type shall not simultaneously indicate high-speed imaging and time-lapse imaging).

**spti\_time\_scale** specifies the number of time units that pass in one second. The value of spti\_time\_scale shall not be equal to 0. For example, a time coordinate system that measures time using a 27 MHz clock has an spti\_time\_scale of 27 000 000.

**spti\_num\_units\_in\_elemental\_interval** specifies the number of time units of a clock operating at the frequency spti\_time\_scale Hz that corresponds to the indicated elemental source picture interval of consecutive pictures in output order in the CLVS.

The indicated elemental source picture interval, also to be denoted by the variable ElementalSourcePictureInterval, in units of seconds, is equal to the quotient of spti\_num\_units\_in\_elemental\_interval divided by spti\_time\_scale. For example, to represent an elemental source picture interval equal to 0.04 seconds, spti\_time\_scale may be equal to 27 000 000 and spti\_num\_units\_in\_elemental\_interval may be equal to 1 080 000.

**spti\_max\_sublayers\_minus\_1** plus 1 specifies the maximum number of temporal sublayers for which picture interval scale factor (spti\_sublayer\_interval\_scale\_factor[ i ]) and synthesized flag (spti\_sublayer\_synthesized\_picture\_flag[ i ]) information is signalled. When spti\_max\_sublayers\_minus\_1 is not present, it is inferred to be equal to TemporalId..

**spti\_sublayer\_interval\_scale\_factor**[ i ], when present, specifies a scale factor used in determining the source picture interval of corresponding consecutive pictures in output order in the CLVS having TemporalId less than or equal to i. The value 0 may be used to indicate that the source picture corresponding to the current decoded output picture is identical to the source picture corresponding to the previous decoded output picture.

The indicated source picture interval associated with output pictures having TemporalId less than or equal to i, denoted by the variable SourcePictureInterval[ i ], in units of seconds, is derived as follows:

SourcePictureInterval[ i ] = ElementalSourcePictureInterval \* spti\_sublayer\_interval\_scale\_factor[ i ] \*  
( 1 − 2 \* temporalReversalFlag ) (8-X)

The variable temporalReversalFlag is equal to ( spti\_source\_type & 0x10 )? 1 : 0.

NOTE 1 – The method of indicating the source picture interval is similar to that used for the timing and HRD parameters syntax used in several video coding standards such as Rec. ITU-T H.266 | ISO/IEC 23090-3, with spti\_time\_scale being similar to that syntax’s time\_scale and spti\_num\_units\_in\_elemental\_interval being similar to that syntax’s num\_units\_in\_tick, and thus the variable ElementalSourcePictureInterval being similar to the variable ClockTick in Rec. ITU-T H.266 | ISO/IEC 23090-3. Since ElementalSourcePictureInterval is multiplied by spti\_sublayer\_interval\_scale\_factor[ i ] when calculating SourcePictureInterval[ i ], it is possible to represent the same value of SourcePictureInterval[ i ] in multiple ways by applying a scale factor to the value of spti\_time\_scale and applying the same scale factor to spti\_num\_units\_in\_elemental\_interval or spti\_sublayer\_interval\_scale\_factor[ i ]. There is no assumption that common scale factors have been removed or that the value of spti\_sublayer\_interval\_scale\_factor[ i ] is equal to 1 for the highest value of i. The reason to allow the same value to be represented in multiple ways is, at least in part, to allow spti\_time\_scale to be chosen to correspond with other timing-related elements used in the system environment, such as the clock rate of 27 MHz used in some multimedia communication systems.

**spti\_sublayer\_synthesized\_picture\_flag**[ i ], when present, equal to 1 indicates that decoded output pictures belonging to the ith temporal sublayer are synthesized and do not correspond to unmodified original source pictures. spti\_sublayer\_synthesized\_picture\_flag[ i ] equal to 0 provides no such indication. When not present, the value of spti\_sublayer\_synthesized\_picture\_flag[ i ] is inferred to be equal to 0.

NOTE 2 – When the TemporalId of an SPTI SEI message is greater than 0, and the SPTI SEI message persists for one or more pictures with lower TemporalId, an encoder can repeat the information of the SPTI SEI message by including it in one or more SPTI SEI messages with lower TemporalId, in order to avoid loss of information when pictures in temporal sublayer(s) are lost or removed.

*Text to be added to a future edition or ammendment of Recommendation ITU-T H.266 | International Standard ISO/IEC 23090-3:*

**Use of the source picture timing SEI message in VVC**

For purposes of interpretation of the source picture timing SEI message, the following variable is specified:

– TemporalId is set equal to TemporalId.

*Text to be added to a future edition or ammendment of Recommendation ITU-T H.265 | International Standard ISO/IEC 23008-2:*

**Use of the source picture timing SEI message in HEVC**

For purposes of interpretation of the source picture timing SEI message, the following variable is specified:

– TemporalId is set equal to TemporalId.

*Text to be added to a future edition or ammendment of Recommendation ITU-T H.264 | International Standard ISO/IEC 14496-10:*

**Use of the source picture timing SEI message in AVC**

For purposes of interpretation of the source picture timing SEI message, the following variable is specified:

– If the bitstream conforms to any of the profiles defined in Annex G, H, I, or J, TemporalId is set equal to temporal\_id.

– Otherwise (the bitstream conforms to a profile defined in Annex A), TemporalId is set equal to ( nal\_ref\_idc = = 0 ? 1 : 0 ).

*Add VSEI subclause 8.32*

Modify text of JVET-AE2032 subclause 8.32 per JVET-AF0088 as follows

* 1. **Object mask information SEI message**
     1. **Object mask information SEI message syntax**

|  |  |
| --- | --- |
| object\_mask\_info( payloadSize ) { | **Descriptor** |
| **omi\_cancel\_flag** | u(1) |
| if( !om\_cancel\_flag ) { |  |
| **omi\_aux\_id\_minus128** | ue(v) |
| **omi\_num\_primary\_pic\_layer\_minus1** | ue(v) |
| for( i = 0; i <= omi\_num\_primary\_pic\_layer\_minus1; i++ ) |  |
| **omi\_primary\_pic\_layer\_id**[ i ] | ue(v) |
| **omi\_mask\_id\_length\_minus8** | ue(v) |
| **omi\_mask\_confidence\_info\_present\_flag** | u(1) |
| if( omi\_mask\_confidence\_info\_present\_flag ) |  |
| **omi\_mask\_confidence\_length\_minus1** | u(4) |
| **omi\_mask\_depth\_info\_present\_flag** | u(1) |
| if( omi\_mask\_depth\_info\_present\_flag ) |  |
| **omi\_mask\_depth\_length\_minus1** | u(4) |
| **omi\_mask\_label\_info\_present\_flag** | u(1) |
| if( omi\_mask\_label\_info\_present\_flag ) { |  |
| **omi\_mask\_label\_language\_present\_flag** | u(1) |
| if( omi\_mask\_label\_language\_present\_flag ) { |  |
| while( !byte\_aligned( ) ) |  |
| **omi\_bit\_equal\_to\_zero** | f(1) |
| **omi\_mask\_lable\_language** | st(v) |
| } |  |
| } |  |
| for( i = 0; i <= omi\_num\_primary\_pic\_layer\_minus1; i++ ) |  |
| for( j = 0; j < numAuxLayer[ omi\_primary\_pic\_layer\_id[ i ] ]; j++ ) { |  |
| **omi\_mask\_pic\_update\_flag**[ i ][ j ] | f(1) |
| if( omi\_mask\_pic\_update\_flag[ i ][ j ] ) { |  |
| **omi\_num\_mask\_in\_pic\_update**[ i ][ j ] | ue(v) |
| for( k = 0; k < omi\_num\_mask\_in\_pic\_update[ i ][ j ]; k++ ) { |  |
| **omi\_mask\_id**[ i ][ j ][ k ] | u(v) |
| **omi\_mask\_bounding\_box\_present\_flag**[ i ][ j ][ k ] | u(1) |
| if( omi\_mask\_bounding\_box\_present\_flag[ i ][ j ][ k ] ) { |  |
| **omi\_mask\_top**[ i ][ j ][ k ] | u(16) |
| **omi\_mask\_left**[ i ][ j ][ k ] | u(16) |
| **omi\_mask\_width**[ i ][ j ][ k ] | u(16) |
| **omi\_mask\_height**[ i ][ j ][ k ] | u(16) |
| } |  |
| **omi\_mask\_cancel**[ i ][ j ][ k ] | u(1) |
| if( !omi\_mask\_cancel[ i ][ j ][ k ] ) { |  |
| if( omi\_mask\_confidence\_info\_present\_flag ) |  |
| **omi\_mask\_confidence**[ i ][ j ][ k ] | u(v) |
| if( omi\_mask\_depth\_info\_present\_flag ) |  |
| **omi\_mask\_depth**[ i ][ j ][ k ] | u(v) |
| while( !byte\_aligned( ) ) |  |
| **omi\_bit\_equal\_to\_zero** | f(1) |
| if( omi\_mask\_label\_info\_present\_flag ) |  |
| **omi\_mask\_label**[ i ][ j ][ k ] | st(v) |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |
| } |  |

* + 1. **Object mask information information SEI message semantics**

The object mask information (OMI) SEI message provides information about object mask pictures coded as auxiliary pictures. Object mask auxiliary pictures have nuh\_layer\_id equal to sdi\_layer\_id[ i ] and sdi\_aux\_id [ i ] in the range of 128 to 159, inclusive, for any value of i in range of 0 to sid\_max\_layers\_minus1, inclusive.

NOTE 1 – Each object mask auxiliary picture layer is associated with one primary picture layer and one primary picture layer may be associated with one or more object mask auxiliary picture layers.

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 conformance cropping window left offset, ConfWinLeftOffset

– A conformance cropping window top offset, ConfWinTopOffset

– A chroma format indicator, denoted herein by ChromaFormatIdc, as described in clause 7.3.

The variables SubWidthC and SubHeightC are derived from ChromaFormatIdc as specified by Table 2.

When an access unit contains an auxiliary picture picA in a layer, with nuh\_layer\_id equal to nuhLayerIdA, that is indicated as an object mask auxiliary layer by an OMI SEI message, and a primary picture picB in a layer, with nuh\_layer\_id equal to nuhLayerIdB, that is indicated as a primary layer by the OMI SEI message, OMI SEI message persists in output order until one or more of the following conditions are true:

– A CLVS containing the auxiliary picture picA ends.

– A CLVS containing the primary picture picB ends.

– A CVS ends.

– The bitstream ends.

**omi\_cancel\_flag** equal to 1 indicates that the SEI message cancels the persistence of any previous object mask information SEI message in output order that is associated with one or more primary picture layers to which this SEI applies. om\_cancel\_flag equal to 0 indicates that object mask information follows.

**omi\_aux\_id\_minus128** plus 128 indicates the value of sdi\_aux\_id of object mask auxiliary picture layer. om\_aux\_id\_minus128 shall be in the range of 0 to 31, inclusive.

When a CVS does not contain an SDI SEI message with sdi\_aux\_id[ i ] equal to omi\_aux\_id\_minus128 + 128 for at least one value of i, no picture in the CVS shall be associated with an OMI SEI message.

When an AU contains both an SDI SEI message with sdi\_aux\_id[ i ] equal to omi\_aux\_id\_minus128 + 128 for at least one value of i and an OMI SEI message, the SDI SEI message shall precede the OMI SEI message in decoding order.

**omi\_num\_primary\_pic\_layer\_minus1** plus 1indicates the number of primary picture layers associated with the object mask auxiliary picture layers to which this SEI message applies. The value of omi\_num\_primary\_pic\_layer\_minus1 shall be in the range of 0 to sdi\_max\_layers\_minus1.

**omi\_primary\_pic\_layer\_id**[ i ] specifies the nuh\_layer\_id value of the i-th primary picture layer to which this OMI SEI message applies. The value of sdi\_aux\_id[ j ] shall be equal to 0 for any value of j in the range of 0 to sid\_max\_layers\_minus1, inclusive, if sdi\_layer\_id[ j ] equal to omi\_primary\_pic\_layer\_id[ i ]

**omi\_mask\_id\_length\_minus8** plus 8 specifies the length, in bits, of omi\_mask\_id[ i ][ j ][ k ] syntax elements.

**omi\_mask\_confidence\_info\_present\_flag** equal to 1 indicates that omi\_mask\_confidence[ i ][ j ][ k ] syntax elements are present. omi\_mask\_confidence\_info\_present\_flag equal to 0 indicates that omi\_mask\_confidence[ i ][ j ][ k ]syntax elements are not present. It is a requirement of bitstream conformance that the value of omi\_mask\_confidence\_info\_present\_flag shall be the same for all object\_mask\_info( ) syntax structures within a CLVS.

**omi\_mask\_confidence\_length\_minus1** plus 1 specifies the length, in bits, of the omi\_mask\_confidence[ i ][ j ][ k ] syntax elements. It is a requirement of bitstream conformance that the value of omi\_mask\_confidence\_length\_minus1 shall be the same for all object\_mask\_info( ) syntax structures within a CLVS.

**omi\_object\_depth\_info\_present\_flag** equal to 1 indicates that omi\_object\_depth[ i ][ j ][ k ] syntax elements are present. omi\_object\_depth\_info\_present\_flag equal to 0 indicates that omi\_object\_depth[ i ][ j ][ k ]syntax elements are not present. It is a requirement of bitstream conformance that the value of omi\_object\_depth\_info\_present\_flag shall be the same for all object\_mask\_info( ) syntax structures within a CLVS.

**omi\_object\_depth\_length\_minus1** plus 1 specifies the length, in bits, of the omi\_object\_depth[ i ][ j ][ k ] syntax elements. It is a requirement of bitstream conformance that the value of omi\_object\_depth\_length\_minus1 shall be the same for all object\_mask\_info( ) syntax structures within a CLVS.

**omi\_mask\_label\_info\_present\_flag** equal to 1 indicates that omi\_mask\_label\_language\_present\_flag and omi\_mask\_label[ i ][ j ][ k ] syntax elements are present. omi\_mask\_label\_info\_present\_flag equal to 0 indicates that omi\_mask\_label\_language\_present\_flag and omi\_mask\_label[ i ][ j ][ k ] syntax elements are not present.

**omi\_mask\_label\_language\_present\_flag** equal to 1 indicates that omi\_mask\_lable\_language syntax element is present. omi\_mask\_label\_language\_present\_flag equal to 0 indicates that omi\_mask\_lable\_language syntax element is not present.

**omi\_bit\_equal\_to\_zero** shall be equal to 0.

**omi\_mask\_lable\_language** contains a language tag as specified by IETF RFC 5646 followed by a null termination byte equal to 0x00. The length of the omi\_mask\_lable\_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.

**omi\_mask\_pic\_update\_flag**[ i ][ j ] equal to 1 indicates the mask information of j-th object mask auxiliary picture associated with i-th primary picture is signalled. omi\_mask\_pic\_update\_flag[ i ][ j ] equal to 0 indicates the mask information of j-th object mask auxiliary picture associated with i-th primary picture is not signalled. When the mask information of j-th object mask auxiliary picture associated with i-th primary picture is not present, the persistence mechanism is used, that is the information is inherited from the last OMI SEI message which signals the mask information of j-th object mask auxiliary picture associated with i-th primary picture.

**omi\_num\_mask\_in\_pic\_update**[ i ][ j ] indicates the number of object masks of which the information to be signalled in the j-th auxiliary picture associated with i-th primary picture. omi\_num\_mask\_in\_pic\_update [ i ][ j ] shall be in the range of 0 to (1<<BitDepthY) – 1, inclusive, where BitDepthY is the bit depth for the samples of the luma component. The variable omiNumMaskInPic[ i ][ j ] indicating the number of object masks in the j-th auxiliary picture associated with i-th primary picture is set to omi\_num\_mask\_in\_pic\_update[ i ][ j ] when the current SEI message is the first OMI SEI message in the current CLVS.

The variable numAuxLayer[ primaryLayerId ] indicates the number of the auxiliary picture layers associated with primary picture layer with nuh\_layer\_id equal to primaryLayerId. The variable associatedAuxLayerId[ primaryLayerId ][ i ] indicates the value of nuh\_layer\_id of the i-th auxiliary picture layer associated with primary picture layer with nuh\_layer\_id equal to primaryLayerId. numAuxLayer[ primaryLayerId ] and associatedAuxLayerId[ primaryLayerId ][ i ] are derived as follows.

for( i = 0; i <= sdi\_max\_max\_layers\_minus1; i++ )  
 numAuxLayer[ sdi\_layer\_id[ i ] ] = 0;  
for( i = 0; i <= sdi\_max\_layers\_minus1; i++ ) {  
 if( sdi\_aux\_id[ i ] = = omi\_aux\_id\_minus128 + 128 ) {  
 for( j = 0; j <= sdi\_num\_associated\_primary\_layers\_minus1[ i ]; j++ ) { (xx)  
 primaryLayerId = sdi\_layer\_id[ sdi\_associated\_primary\_layer\_idx[ i ][ j ] ];  
 associatedAuxLayerId[ primaryLayerId ][ numAuxLayer[ primaryLayerId ] ] = sdi\_layer\_id[ i ];  
 numAuxLayer[ primaryLayerId ]++;  
 }  
 }  
}

**omi\_mask\_id**[ i ][ j ][ k ] indicates the identifier of k-th object mask in the j-th object mask auxiliary picture associated with the i-th primary picture..

The variable maskId[ i ][ j ][ k ] specifying the object mask identifier of k-th object mask in the j-th object mask auxiliary picture associated with i-th primary picture in the SEI message is derived as follows:

for( i = 0; i <= omi\_num\_primary\_pic\_layer\_minus1; i++ ) {  
 for( j = 0; j < numAuxLayer[omi\_primary\_pic\_layer\_id[ i ]]; j++ ) {   
 for( k = 0; k < omiNumMaskInPic[ i ][ j ]; k++ ) {   
 maskId[ i ][ j ][ k ] = omi\_mask\_id[ i ][ j ][ k ] + (1<<BitDepthY)\*j (xx)  
 }  
 }  
}

**omi\_mask\_bounding\_box\_present\_flag**[ i ][ j ][ k ] equal to1 indicates the syntax elements omi\_mask\_top[ i ][ j ][ k ], omi\_mask\_left[ i ][ j ][ k ], omi\_mask\_width[ i ][ j ][ k ], and omi\_mask\_height[ i ][ j ][ k ], are present. omi\_num\_mask\_in\_pic\_update[ i ][ j ][k] equal to 0 indicates syntax elements, omi\_mask\_top[ i ][ j ][ k ], omi\_mask\_left[ i ][ j ][ k ], omi\_mask\_width[ i ][ j ][ k ], and omi\_mask\_height[ i ][ j ][ k ], are not present.

**omi\_mask\_top**[ i ][ j ][ k ], **omi\_mask\_left**[ i ][ j ][ k ], **omi\_mask\_width**[ i ][ j ][ k ], and **omi\_mask\_height**[ i ][ j ][ k ] indicate the coordinates of the top-left corner and the width and height, respectively, of the bounding box of the object mask with identifier equal to omi\_mask\_id[ i ][ j ][ k ] in the cropped decoded picture, relative to the conformance cropping window specified by the active SPS.

The value of omi\_mask\_left[ i ][ j ][ k ] shall be in the range of 0 to ( CroppedWidth / SubWidthC – 1 ), inclusive, CroppedWidth and SubWidthC being associated to the the j-th object mask auxiliary picture associated with i-th primary picture. When it is not present, the value of omi\_mask\_left[ i ][ j ][ k ] is inferred to be 0.

The value of omi\_mask\_top[ i ][ j ][ k ] shall be in the range of 0 to ( CroppedHeight / SubHeightC – 1 ), inclusive, CroppedHeight  and SubHeightC  being associated to the the j-th object mask auxiliary picture associated with i-th primary picture. When it is not present, the value of omi\_mask\_top[ i ][ j ][ k ] is inferred to be 0.

The value of omi\_mask\_width[ i ][ j ][ k ] shall be in the range of 0 to ( CroppedWidth / SubWidthC − omi\_mask\_left[ i ][ j ][ k ] ), inclusive. When it is not present, the value of omi\_mask\_width [ i ][ j ][ k ] is inferred to be ( CroppedWidth / SubWidthC − omi\_mask\_left[ i ][ j ][ k ] ).

The value of omi\_mask\_height[ i ][ j ][ k ] shall be in the range of 0 to ( CroppedHeight / SubHeightC − omi\_mask\_top[ i ][ j ][ k ] ), inclusive. When it is not present, the value of omi\_mask\_height [ i ][ j ][ k ] is inferred to be ( CroppedHeight / SubWidthC − omi\_mask\_top[ i ][ j ][ k ] ).

The identified object mask is within a bounding box containing luma samples with horizontal coordinates from SubWidthC \* ( ConfWinLeftOffset + omi\_mask\_left[ i ][ j ][ k ] ) to SubWidthC \* ( ConfWinLeftOffset + omi\_mask\_left[ i ][ j ][ k ] + omi\_mask\_width[ i ][ j ][ k ] ) − 1, inclusive, and vertical coordinates from SubHeightC \* ( ConfWinTopOffset + omi\_mask\_top[ i ][ j ][ k ] ) to SubHeightC \* ( ConfWinTopOffset + omi\_mask\_top[ i ][ j ][ k ] + omi\_mask\_height[ i ][ j ][ k ] ) − 1, inclusive.

Variable I[ i ][ j ][ x ][ y ] is the decoded value of the sample at the relative sample location (x, y) in the j-th object mask auxiliary picture associated with the i-th primary picture. The following process is to determine each mask region in each auxiliary picture.

for( i = 0; i <= omi\_num\_primary\_pic\_layer\_minus1; i++ ) {  
 for( j = 0; j < numAuxLayer[omi\_primary\_pic\_layer\_id[ i ]]; j++ ) {   
 for( k = 0; k < omiNumMaskInPic[ i ][ j ]; k++ ) { (xx)  
 if( pI[ i ][ j ][ x ][ y ] == omi\_mask\_id[ i ][ j ][ k ]  
 && x >= omi\_mask\_left[ i ][ j ][ k ]   
 && x < omi\_mask\_left[ i ][ j ][ k ] + omi\_mask\_width[ i ][ j ][ k ]  
 && y >= omi\_mask\_top[ i ][ j ][ k ]   
 && y < omi\_mask\_top[ i ][ j ][ k ] + omi\_mask\_height[ i ][ j ][ k ] )  
 The sample at location (x, y) in the j-th object mask auxiliary picture associated with   
 the i-th primary picture is associated with the object mask with the identifier of  
 maskId[ i ][ j ][ k ]  
 }  
 }  
}

**omi\_mask\_cancel**[ i ][ j ][ k ] equal to 1 cancels the persistence scop of object mask with identifier equal to om\_mask\_id[ i ][ j ][ k ]. omi\_mask\_cancel[ i ][ j ][ k ] equal to 0 indicates the information of object mask with identifier equal to omi\_mask\_id[ i ][ j ] is signalled.

It is a requirement of bitstream conformance that when omi\_mask\_id[ i ][ j ][ k ] with value equal to omiMaskId is parsed for the first time in the current CLVS, the value of the corresponding omi\_mask\_cancel[ i ][ j ][ k ] shall be equal to 0.

**omi\_mask\_confidence**[ i ][ j ][  ] indicates the degree of confidence associated with the k-th object mask in the j-th object mask auxiliary picture associated with i-th primary picture, in units of 2-( omi\_mask\_confidence\_length\_minus1 + 1 ), such that a higher value of omi\_mask\_confidence[ i ][ j ][ k ] indicates a higher degree of confidence. The length of the omi\_mask\_confidence[ i ][ j ][ k ] syntax element is omi\_mask\_confidence\_length\_minus1 + 1 bits.

**omi\_mask\_depth**[ i ][ j ][ k ] indicates the object depth associated with the k-th object mask in the j-th object mask auxiliary picture associated with i-th primary picture. A smaller value of omi\_mask\_depth indicates a shorter distance to the object. The length of the omi\_mask\_depth[ i ][ j ][ k ] syntax element is omi\_object\_depth\_length\_minus1 + 1 bits.

**omi\_mask\_label**[ i ][ j ][ k ] specifies the contents of the label associated with k-th object mask in the j-th object mask auxiliary picture associated with i-th primary picture. The length of the omi\_mask\_label[ i ][ j ][ k ] syntax element shall be less than or equal to 255 bytes, not including the null termination byte.

*Add VSEI subclause 8.33*

Add subclause 8.33 per JVET-AF00141 as follows

8.33 Image format metadata SEI messages

8.33.1 Exif metadata SEI message

8.33.1.1 Exif metadata SEI message syntax

|  |  |
| --- | --- |
| exif\_metadata( payloadSize ) { | Descriptor |
| **exif\_cancel\_flag** | u(1) |
| if( !exif\_cancel\_flag ) { |  |
| **exif\_persistence\_flag** | u(1) |
| **exif\_mode\_id** | u(8) |
| if( exif\_mode\_id = = 0 ) |  |
| for( i = 1; i < payloadSize; i++ ) |  |
| **exif\_data\_payload\_byte** | b(8) |
| else if( exif\_mode\_id = = 1 ) |  |
| **exif\_data\_uri** | st(v) |
| } |  |
| } |  |

8.33.1.2 Exif metadata SEI message semantics

The Exchangeable Image File (Exif) Format for digital still cameras includes a set of metadata that captures information regarding the digital photography process that was used to record an image. Such metadata includes: the acceleration vector of the camera at the time the image was captured, camera lens information, GPS data, the color space used, spectral sensitivity, maximum lens aperture, and more.

Exif metadata is specified by any of the Exchangeable image file (Exif) format for digital still cameras standards developed jointly by the Camera & Imaging Products Association (CIPA) and the Japan Electronics and Information Technology Industries Association (JEITA), for the associated video source pictures prior to encoding, e.g., for camera-captured content. To date, there are multple versions of Exif deployed, each with its own indicator to signal the version number in use.

[Ed. Note: Neither CIPA nor JEITA are listed in the organizations for which normative references may be made to documents published by CIPA or JEITA. The list of such organizations is available at: <https://www.itu.int/net4/ITU-T/lists/sdo.aspx>. To add either CIPA or JEITA to the list, the A.4 process needs to be executed for one or both organizations, and their status as A.5 organizations needs to be approved by SG16. There would probably also need to be a specific list of documents referenced rather than the concept of “any of the” standards produced by those organizations.]

The Exif metadata SEI message specifies an SEI message for which the payload of Exif metadata can be carried in the video bitstream.

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

**exif\_persistence\_flag** specifies the persistence of the Exif metadata SEI message for the current layer.

exif\_persistence\_flag equal to 0 specifies that the Exif metadata SEI message applies to the current decoded picture only.

exif\_persistence\_flag equal to 1 specifies that the Exif metadata 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 an Exif metadata SEI message is output that follows the current picture in output order.

**exif\_mode\_id** specifies the mode by which the Exif metadata is obtained.

– If exif\_mode\_id is equal to 0, the Exif metadata is obtained directly from the payload of the SEI message.

– Otherwise if exif\_mode\_id is equal to 1, the Exif metadata is obtained from a URI with syntax and semantics as specified in IETF Internet Standard 66.

All other values of exif\_mode\_id are reserved.

**exif\_data\_payload\_byte** shall be a byte containing data having syntax and semantics as specified by any of the Exchangeable image file format for digital still cameras standards developed jointly by the Camera & Imaging Products Association (CIPA) and the Japan Electronics and Information Technology Industries Association (JEITA).

**exif\_data\_uri** shall contain a URI with syntax and semantics as specified in IETF Internet Standard 66 identifying the Exif metadata.

8.33. JFIF metadata SEI message

8.33.2.1 JFIF metadata SEI message syntax

|  |  |
| --- | --- |
| jfif\_metadata( payloadSize ) { | Descriptor |
| **jfif\_cancel\_flag** | u(1) |
| **jfif\_type\_id** | u(8) |
| if( !jfif\_cancel\_flag ) { |  |
| **jfif\_persistence\_flag** | u(1) |
| if( jfif\_type\_id = = 0 ) |  |
| for( i = 1; i < payloadSize; i++ ) |  |
| **jfif\_data\_payload\_byte** | b(8) |
| else if( jfif\_type\_id = = 1 ) |  |
| for( i = 1; i < payloadSize; i++ ) |  |
| **jfif\_extension\_payload\_byte** | b(8) |
| else if( jfif\_type\_id = = 2 ) |  |
| for( i = 1; i < payloadSize; i++ ) |  |
| **jfif\_header\_payload\_byte** | b(8) |
| } |  |
| } |  |

[Ed. (GJS): Are some of those grid lines thicker than others?]

8.33.2.2 JFIF metadata SEI message semantics

The JFIF metadata SEI message contains JFIF metadata, as specified by ITU-T Recommendation T.871 | ISO/IEC International Standard 10918-5.

JFIF metadata and its semantics are specified by ITU-T Recommendation T.871 | ISO/IEC International Standard 10918-5 (hereafter, the JFIF standard). Of particular importance is the distinction by JFIF between an APP0 marker that contains information describing the organization of the image data, and an APP0 marker that contains “extension” information. [Ed. Note: Needs formal reference (in clause 2). Does that document define what an “APP0 marker” is?]

A single APP0 marker that carries the information to describe the organization of the image is signalled by the zero-terminated string “JFIF” (‘0x4A46494600’) within that APP0 marker. One or more subsequent APP0 markers may follow the first APP0 marker in which the subsequent APP0 marker(s) carries “extension” information. In particular, the JFIF extension mechanism can be used to carry image thumbnails, although there is no specification by the JFIF standard for the precise type of data carried in JFIF extensions.

Approaches for carriage of JFIF within an SEI message include:

– The JFIF payload is carried entirely as a single “container” within an SEI message. In this approach, the SEI payload includes the payloads from the first and all, if any, subsequent JFIF APP0 markers for an image.

– The JFIF payload from only extension APP0 markers, i.e., all JFIF APP0 markers that appear subsequent to the first APP0 marker, is carried as a single container. This approach facilitates the carriage of thumbnail images without the overhead of the carriage of the first JFIF APP0 marker.

– The JFIF payload from only the first JFIF APP0 marker is carried as a single container.

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

**jfif\_persistence\_flag** specifies the persistence of the JFIF metadata SEI message for the current layer.

jfif\_persistence\_flag equal to 0 specifies that the JFIF metadata SEI message applies to the current decoded picture only.

jfif\_persistence\_flag equal to 1 specifies that the JFIF metadata 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 an JFIF metadata SEI message is output that follows the current picture in output order.

**jfif\_type\_id** specifies the type of JFIF metadata.

– If jfif\_type\_id is equal to 0, the remaining SEI payload bytes shall contain JFIF metadata from one or more concatenated APP0 marker segments without the APP0 marker segment identifiers themselves, i.e., an APP0 marker segment with bytes containing 0x4A46494600, i.e, the zero-terminated string "JFIF", according to Rec. ITU-T T.50 or ISO 646 coding, and one or more APP0 marker segments containing 0x4A46585800, i.e, the zero-terminated string "JFXX", according to Rec. ITU-T T.50 or ISO 646 coding.

– Otherwise if jfif\_type\_id is equal to 1, the remaining SEI payload bytes shall contain bytes from one or more JFIF extension markers, i.e., one or more APP0 markers that include 0x4A46585800, i.e, the zero-terminated string "JFXX", according to Rec. ITU-T T.50 or ISO 646 coding.

– Otherwise if jfif\_type\_id is equal to 2, the remaining SEI payload bytes shall contain bytes obtained from the first APP0 marker following the 0xFFD8 Start of Image marker within the image. The first APP0 marker includes 0x4A46494600, i.e, the zero-terminated string "JFIF", according to Rec. ITU-T T.50 or ISO 646 coding.

All other values of jfif\_type\_id are reserved.

**jfif\_data\_payload\_byte** shall be a byte containing data from one or more concatenated APP0 JPEG marker segments without the bytes 0xFFE0 from each APP0 marker segment identifier, each such marker segment having syntax and semantics specified by ITU-T Recommendation T.871 | ISO/IEC International Standard 10918-5.

**jfif\_extension\_payload\_byte** shall be a byte containing data from one or more concatenated APP0 JPEG marker segments, without the bytes 0xFFE0 from each APP0 marker segment identifier, each such marker segment having syntax and semantics of a JFIF extension APP0 marker segment specified by ITU-T Recommendation T.871 | ISO/IEC International Standard 10918-5.

**jfif\_header\_payload\_byte** shall be a byte containing data from the first APP0 JPEG marker segment following the 0xFFD8 Start of Image marker segment, without the bytes 0xFFE0 from the APP0 marker segment identifier, such marker segment having syntax and semantics of the first APP0 marker segment immediately following the 0xFFD8 Start of Image marker segment specified by ITU-T Recommendation T.871 | ISO/IEC International Standard 10918-5.

8.33.3 XMP metadata SEI message

8.33.3.1 XMP metadata SEI message syntax

|  |  |
| --- | --- |
| xmp\_metadata( payloadSize ) { | Descriptor |
| **xmp\_cancel\_flag** | u(1) |
| if( !xmp\_cancel\_flag ) { |  |
| **xmp\_persistence\_flag** | u(1) |
| for( i = 1; i < payloadSize; i++ ) |  |
| **xmp\_data\_payload\_byte** | b(8) |
| } |  |
| } |  |

8.33.3.1 XMP metadata SEI message semantics

The XMP metadata SEI message specifies an SEI message for XMP metadata.

Extensible Metadata Platform (XMP) metadata is specified in ISO 16684-1: Graphic Technology – Extensible metadata platform (XMP) specification. [Ed. Note (GJS): Formal reference needed in clause 2.] XMP is widely deployed by digital cameras and digital image editing packages to record provenance and editing history with an image. The most common set of metadata included in XMP refers to a vocabulary defined by the Dublin Core Metadata Initiative (DCMI)[[1]](#footnote-1), e.g., to store information such as digital rights ownership and names of software packages used to modify the image.

The XMP metadata SEI message indicates the XMP metadata, as specified by [1] in XML or [3] in JSON. [Ed. Note: Fix or remove unresolved references and undefined abbreviations.]

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

**xmp\_persistence\_flag** specifies the persistence of the XMP metadata SEI message for the current layer.

xmp\_persistence\_flag equal to 0 specifies that the XMP metadata SEI message applies to the current decoded picture only.

xmp\_persistence\_flag equal to 1 specifies that the XMP metadata 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 an XMP metadata SEI message is output that follows the current picture in output order.

**xmp\_data\_payload\_byte** shall be a byte containing data having syntax and semantics as specified by ISO 16684-1: Graphic Technology – Extensible metadata platform (XMP) specification – Part 1 Data model, serialization and core properties or ISO 16684-3: Graphic Technology – Extensible metadata platform (XMP) specification – Part 3 JSON-LD serialization of XMP.

*Add VSEI subclause 8.34*

Add subclause 8.34 per JVET-AF0167 as follows

**8.34 Multiplane image information SEI message**

8.34.1 Multiplane image information SEI message syntax

|  |  |
| --- | --- |
| multiplane\_image\_information( payloadSize ) { | **Descriptor** |
| **mpii\_num\_layers\_minus1** | ue(v) |
| **mpii\_layer\_depth\_equal\_distance\_flag** | u(1) |
| if( mpii\_layer\_depth\_equal\_distance\_flag ) { |  |
| depth\_rep\_info\_element( ZNearSign, ZNearExp, ZNearMantissa, ZNearManLen ) |  |
| depth\_rep\_info\_element( ZFarSign, ZFarExp, ZFarMantissa, ZFarManLen ) |  |
| } else |  |
| for( i = 0; i <= mpii\_num\_layer\_minus1; i++ ) |  |
| depth\_rep\_info\_element( ZSign[ i ], ZExp[ i ], ZMantissa[ i ], ZManLen[ i ] ) |  |
| **mpii\_texture\_opacity\_interleave\_flag** | u(1) |
| if( mpii\_texture\_opacity\_interleave\_flag = = 0 ) |  |
| **mpii\_texture\_opacity\_arrangement\_flag** /\* 0: Top-and-Bottom, 1: Side-by-Side \*/ | u(1) |
| **mpii\_picture\_num\_layers\_in\_height\_minus1** | ue(v) |
| } |  |

8.34.2 Muliplane image information SEI message semantics

The multiplane image information (MPII) SEI message specifies multiplane image (MPI) scene representation information that may be used for view synthesis.

When an MPII SEI message is present in any AU of a CLVS, an MPII SEI message shall be present in the first AU of the CLVS 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 an MPII SEI message is output that follows the current picture in output order.

NOTE 1 – This SEI message can work together with the multiview acquisition information SEI message for view synthesis. The multiview acquisition information SEI message specifies the intrinsic and extrinsic parameters for current camera view. When multiple views are available, the reconstructed novel views can be rendered from multiplane images of nearby views.

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

– Cropped decoded output picture width and height in units of luma samples, denoted herein by CroppedWidth and CroppedHeight, respectively.

– A chroma format indicator, denoted herein by ChromaFormatIdc, as described in subclause ‎7.3.

– A cropped decoded picture array decPicCurr0[ cIdx ][ x ][ y ], with cIdx = 0..(ChromaFormatIdc = = 0 ) ? 0 : 2, x = 0..( cIdx = = 0 ) ? CroppedWidth : CroppedWidth / SubWidthC − 1, y = 0..( cIdx = = 0 ) ? CroppedHeight : CroppedHeight / SubHeightC − 1.

– In output order a temporally following cropped decoded picture array decPicCurr1[ cIdx ][ x ][ y ], with cIdx = 0..(ChromaFormatIdc = = 0 ) ? 0 : 2, x = 0..( cIdx = = 0 ) ? CroppedWidth : CroppedWidth / SubWidthC − 1, y = 0..( cIdx = = 0 ) ? CroppedHeight : CroppedHeight / SubHeightC − 1.

The variables SubWidthC and SubHeightC are derived from ChromaFormatIdc as specified by Table 2.

**mpii\_num\_layers\_minus1** plus 1 specifies the number of texture and opacity layers for the MPI representation.

**mpii\_layer\_depth\_equal\_distance\_flag** equal to 1 indicates that equal distances are used to generate MPI layers and depth parameters for each layer. In this case, Z[ i ] can be derived using the nearest depth value ZNear and the farthest depth value ZFar.

The depth value for i-th MPI layer, Z[ i ], is derived as follows:

Z[ i ] = i \* ( ZFar − Znear ) ÷ (mpi\_num\_layers\_minus1 ) + ZNear (xx)

mpii\_layer\_depth\_equal\_distance\_flag equal to 0 indicates that the depth information for each layer follows next in the SEI message.

The variables in the x column of Table xx are derived from the respective variables in the s, e, n and v columns of Table xx as follows:

– If the value of e is in the range of 0 to 127, exclusive, x is set equal to ( − 1)s \* 2( e − 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 xx – Association between depth parameter variables and syntax elements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x** | **s** | **e** | **n** | **v** |
| ZNear | ZNearSign | ZNearExp | ZNearMantissa | ZNearManLen |
| ZFar | ZFarSign | ZFarExp | ZFarMantissa | ZFarManLen |
| Z[ i ] | ZSign[ i ] | ZExp | ZMantissa | ZManLen |

NOTE 3 – In some applications, disparity is used instead of depth (the disparity value D and depth value Z relationship is D = 1 ÷ Z). Correspoding to Equation (xx), the disparity value for the i-th MPI layer is D[ i ] = i \* ( DFar − Dnear ) ÷ (mpi\_num\_layers\_minus1 ) + DNear.

**mpii\_texture\_opacity\_interleave\_flag** equal to 1 indicates decoded output pictures correspond to temporally interleaved texture and opacity constituent pictures in output order as illustrated in Figure XX. mpii\_texture\_opacity\_interleave\_flag equal to 0 indicates decoded output pictures correspond to spatially packed texture and opacity constituent pictures as illustrated in Figures XX and XX.

**mpii\_texture\_opacity\_arrangement\_flag** equal to 0 indicates decoded output pictures represent texture and opacity constituent pictures in a top-bottom packing arrangement as illustrated in Figure XX. mpii\_texture\_opacity\_arrangement\_flagequal to 1 indicates decoded output pictures represent texture and opacity constituent pictures in a side-by-side packing arrangement as illustrated in Figure XX.

For each specified picture packing arrangement scheme, there are two constituent pictures that are referred to as picture 0 and picture 1. When mpii\_texture\_opacity\_interleave\_flag is equal to 0, the constituent picture associated with the upper-left sample of the decoded picture is considered to be constituent picture 0 and the other constituent picture is considered to be constituent picture 1. When mpii\_texture\_opacity\_interleave\_flag is equal to 1, the first decoded picture in the current CLVS is constituent picture 0 and the next decoded picture in output order is constituent picture 1 and the display time of the constituent picture 0 should be delayed to coincide with the display time of constituent picture 1. The two constituent pictures form the spatially packed texture and opacity picture of a MPI, with picture 0 being associated with the spatially packed texture picture and picture 1 being associated with the spatially packed opacity picture.

**mpii\_picture\_num\_layers\_in\_height\_minus1** plus 1 specifies the number of spatially packed layers in height for picture 0 and picture 1. The variable hLayers is set equal to mpii\_picture\_num\_layers\_in\_height\_minus1 + 1 and the variable wLayers is set equal to (mpii\_num\_layers\_minus1 + 1) / hLayers.

Let variable fWidth and fHeight specify the width and height of picture 0 and picture 1, respectively, and are derived as follows:

– If mpii\_texture\_opacity\_interleave\_flag is equal to 1, the following applies:

fWidth = CroppedWidth

fHeight = CroppedHeight

– Otherwise (mpii\_texture\_opacity\_interleave\_flag is equal to 0)

– If mpii\_texture\_opacity\_arrangement\_flag is equal to 0, the following applies:

fWidth = CroppedWidth , fHeight = CroppedHeight / 2

– Otherwise (mpii\_texture\_opacity\_arrangement\_flag is equal to 1), the following applies:

fWidth = CroppedWidth / 2 , fHeight = CroppedHeight

Let variable cWidth = fWidth / subWidthC and variable cHeight = fHeight / subHeightC.

Let array picture0[ cIdx ][ x ][ y ] specify samples in picture 0 and array picture1[ cIdx ][ x ][ y ] specify samples in picture 1, with cIdx = 0..(ChromaFormatIdc = = 0 ) ? 0 : 2, x = 0..( cIdx = = 0 ) ? fWidth: cWidth − 1, y = 0..( cIdx = = 0 ) ? fHeight : cHeight − 1 and are derived as follows:

– If mpii\_texture\_opacity\_interleave\_flag is equal to 1, the following applies:

picture0[ cIdx ][ x ][ y ] = decPicCurr0[ cIdx ][ x ][ y ]

picture1[ cIdx ][ x ][ y ] = decPicCurr1[ cIdx ][ x ][ y ]

– Otherwise (mpii\_texture\_opacity\_interleave\_flag is equal to 0)

– Let variable cW = ( cIdx = = 0 )? fWidth : cWidth

– Let variable cH = ( cIdx = = 0 )? fHeight : cHeight

– If mpii\_texture\_opacity\_arrangement\_flag is equal to 0, the following applies:

picture0[ cIdx ][ x ][ y ] = decPicCurr0[ cIdx ][ x ][ y ]

picture1[ cIdx ][ x ][ y ] = decPicCurr0[ cIdx ][ x ][ y + cH ]

– Otherwise (mpii\_texture\_opacity\_arrangement\_flag is equal to 1), the following applies:

picture0[ cIdx ][ x ][ y ] = decPicCurr0[ cIdx ][ x ][ y ]

picture1[ cIdx ][ x ][ y ] = decPicCurr0[ cIdx ][ x + cW ][ y ]

Let variable layerWidth and layerHeight specify the width and height for decoded MPI layer, respectively. The variables are derived as follows:

layerWidth = fWidth / wLayers

layerHeight = fHeight / hLayers

The reconstruction of MPI process is described as follows:

The outputs of this process are:

– a 4D MPI texture layer array recTextureLayer[ i ][ cIdx ][ w ][ h ] with i = 0..mpii\_num\_layers\_minus1, cIdx = 0..(ChromaFormatIdc = = 0 ) ? 0 : 2, w = 0..( cIdx = = 0 ) ? layerWidth : layerWidth / SubWidthC − 1, and h = 0..( cIdx = = 0 ) ? layerHeight : layerHeight / SubHeightC − 1.

– a 3D MPI opacity layer array recOpacityLayer[ i ][ w ][ h ] with i = 0..mpii\_num\_layers\_minus1, x = 0..layerWidth − 1, and y = 0..layerHeight − 1.

The array recTextureLayer and array recOpacityLayer are derived as follows:

for( i = 0; i  <=  mpii\_num\_layers\_minus1; i++ ) {  
 k = i % wLayers  
 m = ( i − k ) / hLayers  
 for( cIdx = 0; cIdx < ChromaFormatIdc = = 0 ) ? 1 : 3; cIdx++ )  
 for( h = 0; h < ( cIdx = = 0 ) ? layerHeight : layerHeight / SubHeightC ; h++ )  
 for( w = 0; w < ( cIdx = = 0 ) ? layerWidth : layerWidth / SubWidthC ; w++ ) {  
 u = k \* ( cIdx  = =  0 ) ? layerWidth : layerWidth / SubWidthC + w v = m \* ( cIdx  = =  0 ) ? layerHeight : layerHeight / SubHeightC + h  
 recTextureLayer[ i ][ cIdx ][ w ][ h ] = picture0[ cIdx ][ u ][ v ]  
 }  
 for( h = 0; h < layerHeight; h++ )  
 for( w = 0; w < layerWidth; w++ )  
 recOpacityLayer[ i ][ w ][ h ] = picture1[ 0 ][ k \* layerWidth + w ][ m \* layerHeight + h ]  
}

1. https://www.dublincore.org [↑](#footnote-ref-1)