



DPC: 25/30498957 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000

Fax: +44 (0)20 8996 7400

www.bsigroup.com

Date: 01 July 2025

Origin: International

Latest date for receipt of comments: 24 August 2025

Project No. 2024/02348

Responsible committee: IST/37 Coding of picture, audio, multimedia and hypermedia information

Interested committees:

Title: Draft BS ISO/IEC 23002-7 :2024/Amd 1 Information technology - MPEG video technologies. Part 7: Versatile supplemental enhancement information messages for coded video bitstreams. Amendment 1: Additional SEI messages.

Please notify admin.start@bsigroup.com if you are aware of any keywords that might assist in clarifying or identifying the standard or of the content of this standard:

- i) has any issues related to 3rd party IPR, patent or copyright
- ii) affects other national standard(s)
- iii) requires additional national guidance or information

WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD. THIS DRAFT IS NOT CURRENT BEYOND 24 August 2025

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop <http://shop.bsigroup.com> or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from <https://standardsdevelopment.bsigroup.com/>.

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at <https://standardsdevelopment.bsigroup.com/>. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx
------------------	------------------------

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex (e.g. 3.1)	Paragraph/ Figure/Table/Note	Type of comment	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	



DRAFT Amendment

ISO/IEC 23002-7:2024/ DAM 1

Information technology — MPEG video technologies —

Part 7: Versatile supplemental enhancement information messages for coded video bitstreams

AMENDMENT 1: Additional SEI messages

ICS: 35.040.40

ISO/IEC JTC 1/SC 29

Secretariat: JISC

Voting begins on:
2025-07-01

Voting terminates on:
2025-09-23

This document is circulated as received from the committee secretariat.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

Reference number
ISO/IEC 23002-7:2024/DAM 1:2025(en)

© ISO/IEC 2025  **Documents**
Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

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

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

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

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

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

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

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

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

Information technology — MPEG video technologies —

Part 7:

Versatile supplemental enhancement information messages for coded video bitstreams

AMENDMENT 1: Additional SEI messages

Replace the content of Clause 2 with the following:

- Recommendation ITU-T T.35 (in force), Procedure for the allocation of ITU-T defined codes for non standard facilities.
- IETF RFC 1321 (in force), The MD5 Message-Digest Algorithm.
- IETF RFC 4151 (in force), The 'tag' URI Scheme.
- IETF RFC 5646 (in force), Tags for Identifying Languages.
- IETF Standard 66 (in force), Uniform Resource Identifiers (URI): Generic Syntax.
- ISO/CIE 11664-1 (in force), Colorimetry – Part 1: CIE standard colorimetric observers.
- ISO/IEC 10646 (in force), Information technology – Universal coded character set (UCS).
- ISO/IEC 11578:1996, Information technology – Open Systems Interconnection – Remote Procedure Call (RPC).
- ISO/IEC 15938-17 (in force), Information technology – Multimedia content description interface – Part 17: Compression of neural networks for multimedia content description and analysis
- Doc. CTA-766-D (ANSI), CTA, U.S. and Canadian Region Rating Tables (RRT) and Content Advisory Descriptors for Transport of Content Advisory Information Using ATSC Program and System Information Protocol (PSIP), Consumer Technology Association, Arlington, VA, December 11, 2013
- ISO 20473:2007, Optics and photonics – Spectral bands.
- NIST FIPS PUB 180-4, Secure Hash Standard (SHS).
- ISO/IEC 21617-1:2024, Information technology - JPEG Trust - Part 1: Core Foundation.
- Rec. ITU-T T.871 | ISO/IEC 10918-5 (in force), Information technology - Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF)
- Exchangeable image file format for digital still cameras: Exif Version 3.0, Camera & Imaging Products Association, May 2023.
- ISO 16684-1: Graphic Technology – Extensible metadata platform (XMP) specification– Part 1 Data model, serialization and core properties.
- ISO 16684-3: Graphic Technology – Extensible metadata platform (XMP) specification – Part 3 JSON-LD serialization of XMP.

- ISO 15076-1 Image technology colour management — Architecture, profile format and data structure — Part 1: Based on ICC.1:2010
- ISO 20677 Image technology colour management — Extensions to architecture, profile format and data structure
- TIFF specification rev. 6.0, <https://www.itu.int/itudoc/itu-t/com16/tiff-fx/docs/tiff6.pdf>, June 1992

Replace the definition of IRAP picture in subclause 3.32 with the following:

3.32

IRAP picture

Coded picture starting from which all pictures in the same layer in both decoding order and output order can be decoded without first decoding any picture in the same layer earlier in decoding order in the coded video bitstream, such that when starting from the coded picture for all pictures in the same layer in both decoding order and output order the cropped decoded pictures are fully correct

In clause 3, add the following definitions:

3.71

corresponding picture

picture in a picture list for a particular picture picA that is either picA itself or a processed version of picA generated when the process implied by an SEI message is applied for a particular picture picA

Note 1 to entry: The particular picture picA could be a picture that is not in the picture list, in which case the corresponding picture in the picture list is a processed version of picA. When picA is in the picture list, it's corresponding picture in the picture list is itself.

3.72

inserted picture

picture that was interpolated or extrapolated when the process implied by an SEI message (e.g., an NNPF SEI message activating an NNPF with PictureRateUpsamplingFlag or TemporalExtrapolationFlag equal to 1) is applied

3.73

associated inserted picture

picture picB in the picture list for a particular picture picA that is the corresponding picture of an inserted picture generated when applying the process implied by an SEI message to a corresponding picture of picA or to an associated inserted picture of picA.

Note 1 to entry: The particular picture picA could be a picture that is not in the picture list.

3.74

gradual decoding refresh (GDR) picture

coded picture starting from which all pictures in the same layer in both decoding order and output order can be decoded without first decoding any picture in the same layer earlier in decoding order in the coded video bitstream, such that when starting from a subsequent picture in decoding order for all pictures in the same layer in both decoding order and output order the cropped decoded pictures are fully correct

3.75

gradual decoding refresh (GDR) AU

AU in which each coded picture is a GDR picture

In clause 4, add the following abbreviation:

FGC Film Grain Characteristics

Replace paragraphs 2 through 4 of subclause 6.1 with the following:

Technical specifications that reference this document for carrying SEI messages shall specify a container to carry the payload syntax of each specified SEI message, to identify which SEI message is conveyed through a payloadType variable that indicates the SEI message payload type, and to identify the length in bits of the SEI message payload syntax structure. An example of the container and an example of the SEI message payload syntax structure are the sei_message() syntax structure and the sei_payload() syntax structure, respectively, specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 and Rec. ITU-T H.265 | ISO/IEC 23008-2. The design of the container should provide the ability to detect the number of bits in an SEI message and to allow the number of bits to be increased in future versions of this document, thus enabling this document to provide extensibility by directly appending additional syntax elements to the end of the SEI message payload syntax structure in future versions of this document. The syntax of the container of SEI messages as well as the method of identifying which SEI message is outside the scope of this document.

The length of the VUI parameters syntax structure or an SEI message payload syntax structure in bits is referred to herein by the variable PayloadBits, which is provided by an external means not specified in this document. The number of bytes that contains the payload data is referred to herein by the variable payloadSize, where payloadSize is equal to $\text{Ceil}(\text{PayloadBits} \div 8)$.

Replace the semantics of vui_matrix_coeffs in subclause 7.3 with the following:

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, 16, or 17 unless both of the following conditions are true:

- BitDepth_C is equal to BitDepth_Y .
- ChromaFormatIdc is equal to 3 (the 4:4:4 chroma format).

The specification of the use of vui_matrix_coeffs equal to 0, 16, or 17 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:

- BitDepth_C is equal to BitDepth_Y ,
- BitDepth_C is equal to $\text{BitDepth}_Y + 1$ 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 document).

Replace the last bullet item in the semantics of vui_chroma_sample_loc_type_frame, vui_chroma_sample_loc_type_top_field, and vui_chroma_sample_loc_type_bottom_field in subclause 7.3 with the following:

- Otherwise (ChromaFormatIdc is not equal to 1), the values of the syntax elements vui_chroma_sample_loc_type_frame, vui_chroma_sample_loc_type_top_field and vui_chroma_sample_loc_type_bottom_field shall be ignored.

Modify subclause 8.1 as follows:

Replace Table 4 with the following:

Table 4 — Persistence scope of SEI messages (informative)

SEI message	Persistence scope
Filler payload	The PU containing the SEI message
User data registered by Rec. ITU-T T.35	Unspecified
User data unregistered	Unspecified
Film grain characteristics	Specified by the syntax of the SEI message
Frame packing arrangement	Specified by the syntax of the SEI message
Parameter sets inclusion indication	The CLVS containing the SEI message
Decoded picture hash	The PU containing the SEI message
Mastering display colour volume	The CLVS containing the SEI message
Content light level information	The CLVS containing the SEI message
DRAP indication	The picture associated with the SEI message
Alternative transfer characteristics	The CLVS containing the SEI message
Ambient viewing environment	The CLVS containing the SEI message
Content colour volume	Specified by the syntax of the SEI message
Equirectangular projection	Specified by the syntax of the SEI message
Generalized cubemap projection	Specified by the syntax of the SEI message
Sphere rotation	Specified by the syntax of the SEI message
Region-wise packing	Specified by the syntax of the SEI message
Omnidirectional viewport	Specified by the syntax of the SEI message
Frame-field information	The PU containing the SEI message
Sample aspect ratio information	Specified by the syntax of the SEI message
Annotated regions	Specified by the syntax of the SEI message
Scalability dimension information	The CVS containing the SEI message
Multiview acquisition information	The CVS containing the SEI message
Multiview view position	The CVS containing the SEI message
Depth representation information	Specified by the semantics of the SEI message
Alpha channel information	Specified by the syntax of the SEI message
Extended DRAP indication	The picture associated with the SEI message
Display orientation	Specified by the syntax of the SEI message
Colour transform information	Specified by the syntax of the SEI message
Shutter interval information	The CLVS containing the SEI message
Neural-network post-filter characteristics	The CLVS containing the SEI message
Neural-network post-filter activation	Specified by the syntax of the SEI message
Phase indication	Specified by the semantics of the SEI message
SEI processing order	For each value of po_id, the number of SEI messages and the payload-Type codes of the SEI messages indicated within the SEI processing order SEI message persist for the CLVS containing the SEI processing order SEI message.
Processing order nesting	Depending on the processing-order-nested SEI messages. Each processing-order-nested SEI message has the same persistence scope as if the SEI message was not nested.
Encoder optimization information	Specified by the syntax of the SEI message
Source picture timing information	Specified by the syntax of the SEI message
Object mask information	Specified by the syntax of the SEI message
Modality information	Specified by the syntax of the SEI message
Text description information	Specified by the syntax of the SEI message

Table 4 (continued)

SEI message	Persistence scope
Generative face video	Specified by the semantics of the SEI message
Generative face video enhancement	Specified by the semantics of the SEI message
Digitally signed content initialization	Specified by the semantics of the SEI message
Digitally signed content selection	The PU containing the SEI message
Digitally signed content verification	Specified by the semantics of the SEI message
AI usage restrictions	Specified by the syntax of the SEI message
Packed regions information	Specified by the syntax of the SEI message
Image format metadata	Specified by the syntax of the SEI message

Renumber the existin NOTE as NOTE 1.

Add the following NOTE 2:

NOTE 2 When multiple NNPFs specified by NNPF SEI messages with different nnpfc_purpose values are activated for the same picture, it is possible that the corresponding NNPF processes have some conflict. Similarly, when an SEI message implying a non-NNPF post-processing process is associated with a picture and an NNPF specified by an NNPF SEI message with a related purpose is activated for the same picture, it is possible that the corresponding processes have some conflict. Particular care is expected to be exercised in the design of encoders that generate such multiple SEI messages. For example, process-implying SEI messages that have some conflict can be included in different processing chains defined by SEI processing order SEI messages. Alternatively, multiple process-implying SEI messages with related post-processing operations can be used in contexts in which conflicts of usage are unimportant, or not possible, or are managed – e.g., defined or managed in the controlling application or transport specification, or by controlling the environment in which bitstreams are distributed.

Replace subclause 8.5.1 with the following:

film_grain_characteristics(payloadSize) {	Descriptor
fg_characteristics_cancel_flag	u(1)
if(!fg_characteristics_cancel_flag) {	
fg_model_id	u(2)
fg_separate_colour_description_present_flag	u(1)
if(fg_separate_colour_description_present_flag) {	
fg_bit_depth_luma_minus8	u(3)
fg_bit_depth_chroma_minus8	u(3)
fg_full_range_flag	u(1)
fg_colour_primaries	u(8)
fg_transfer_characteristics	u(8)
fg_matrix_coeffs	u(8)
}	
fg_blending_mode_id	u(2)
fg_log2_scale_factor	u(4)
for(c = 0; c < 3; c++)	
fg_comp_model_present_flag[c]	u(1)
for(c = 0; c < 3; c++)	
if(fg_comp_model_present_flag[c]) {	
fg_num_intensity_intervals_minus1[c]	u(8)

fg_num_model_values_minus1[c]	u(3)
for(i = 0; i <= fg_num_intensity_intervals_minus1[c]; i++) {	
fg_intensity_interval_lower_bound[c][i]	u(8)
fg_intensity_interval_upper_bound[c][i]	u(8)
for(j = 0; j <= fg_num_model_values_minus1[c]; j++)	
fg_comp_model_value[c][i][j]	se(v)
}	
}	
fg_characteristics_persistence_flag	u(1)
}	
if(more_data_in_payload() && payload_extension_present()) {	
SeiExtensionBitsPresentFlag = 1	
fg_spatial_resolution_present_flag	u(1)
if(fg_spatial_resolution_present_flag) {	
fg_pic_width_in_luma_samples	ue(v)
fg_pic_height_in_luma_samples	ue(v)
}	
}	
}	

Replace subclause 8.5.2 with the following:

This SEI message provides the decoder with a parameterized model for a film grain synthesis process. The film grain synthesis process should be applied to the decoded pictures prior to their display.

NOTE 1 For example, an encoder could use the film grain characteristics SEI message to characterize film grain that was present in the original source video material and was removed by pre-processing filtering techniques. Synthesis of simulated film grain on the input images, which could be the decoded pictures or converted from the decoded pictures, for the display process is optional and does not necessarily exactly follow the specified semantics of the film grain characteristics SEI message. When synthesis of simulated film grain on the input images for the display process is performed, there is no requirement that the method by which the synthesis is performed be the same as the parameterized model for the film grain as provided in the film grain characteristics SEI message.

NOTE 2 The display process is not specified in this document.

NOTE 3 SMPTE RDD 5 (2006) specifies a film grain simulator based on the information provided in the film grain characteristics SEI message.

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

- A picture width and picture height in units of luma samples, denoted herein by `PicWidthInLumaSamples` and `PicHeightInLumaSamples`, respectively.
- An input picture `InPic[ c ]` where $c = 0..2$, to which the film grain characteristics SEI message applies.
- When the syntax element `fg_separate_colour_description_present_flag` of the film grain characteristics SEI message is equal to 0, the following additional variables:
 - A chroma format indicator, denoted herein by `ChromaFormatIdc`, as described in subclause 7.3.
 - 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`.

The variables `fgPicWidth` and `fgPicHeight` are derived as follows:

- If `fg_pic_width_in_luma_samples` and `fg_pic_height_in_luma_samples` are not present, `fgPicWidth` and `fgPicHeight` are set equal to `PicWidthInLumaSamples` and `PicHeightInLumaSamples`, respectively.
- Otherwise, `fgPicWidth` and `fgPicHeight` are set equal to `fg_pic_width_in_luma_samples` and `fg_pic_height_in_luma_samples`, respectively.

The film grain models specified in the film grain characteristics SEI message are expressed for application to pictures that have 4:4:4 colour format and a width of `fgPicWidth` and height of `fgPicHeight` with luma and chroma bit depths corresponding to the luma and chroma bit depths of the film grain model and use the same colour representation domain as the identified film grain model. When the colour format of `InPic` is not 4:4:4 or `InPic` uses a different width or height or a different luma or chroma bit depth from that of the film grain model or uses a different colour representation domain from that of the identified film grain model, an unspecified conversion process is expected to be applied to convert `InPic` to the form that is expressed for application of the film grain model.

NOTE 4 Because the use of a specific method is not required for performing the film grain generation function used by the display process, a decoder could, if desired, down-convert the model information for chroma in order to simulate film grain for other chroma formats (4:2:0 or 4:2:2) rather than up-converting the decoded video (using a method not specified in this document) before performing film grain generation.

`fg_characteristics_cancel_flag` equal to 1 indicates that the SEI message cancels the persistence of any previous film grain characteristics SEI message in output order that applies to the current layer. `fg_characteristics_cancel_flag` equal to 0 indicates that film grain modelling information follows.

`fg_model_id` identifies the film grain simulation model as specified in Table 5. The value of `fg_model_id` shall be in the range of 0 to 1, inclusive. The values of 2 and 3 for `fg_model_id` are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders shall ignore film grain characteristics SEI messages with `fg_model_id` equal to 2 or 3.

Table 1 — `fg_model_id` values

Value	Description
0	Frequency filtering
1	Auto-regression

`fg_separate_colour_description_present_flag` equal to 1 indicates that a distinct combination of luma bit depth, chroma bit depth, video full range flag, colour primaries, transfer characteristics, and matrix coefficients for the film grain characteristics specified in the SEI message is present in the film grain characteristics SEI message syntax. `fg_separate_colour_description_present_flag` equal to 0 indicates that the combination of luma bit depth, chroma bit depth, video full range flag, colour primaries, transfer characteristics, and matrix coefficients for the film grain characteristics specified in the SEI message are the same as indicated in VUI parameters for the CLVS.

NOTE 5 When `fg_separate_colour_description_present_flag` is equal to 1, any of the luma bit depth, chroma bit depth, video full range flag, colour primaries, transfer characteristics, and matrix coefficients specified for the film grain characteristics specified in the SEI message could differ from that for the pictures in the CLVS.

When VUI parameters are not present for the CLVS or the value of `vui_colour_description_present_flag` is equal to 0, and equivalent information to that conveyed when `vui_colour_description_present_flag` is equal to 1 is not conveyed by external means, `fg_separate_colour_description_present_flag` shall be equal to 1.

The input image $\hat{I}$, which may be the decoded picture or converted from the decoded picture, used in the pseudocode in this clause is in the same colour representation domain as the simulated film grain signal. Therefore, when any of these parameters does differ from that for the pictures in CLVS, the input image $\hat{I}$ used in the pseudocode in this clause would be in a different colour representation domain than that for the pictures in the CLVS. For example, when the value of `fg_bit_depth_luma_minus8 + 8` is greater than `BitDepthY` (i.e., the bit depth of the luma component of the pictures in the CLVS), the bit depth of the input image $\hat{I}$ used in the pseudocode in this clause is also greater than `BitDepthY`. In such a case, the input image $\hat{I}$ would be generated by converting the actual decoded picture to be in the same colour representation domain as the

simulated film grain signal. The process for converting an actual decoded picture to the 4:4:4 colour format with same colour representation domain as the simulated film grain signal is not specified in this document.

fg_bit_depth_luma_minus8 plus 8 specifies the bit depth used for the luma component of the film grain characteristics specified in the SEI message. When **fg_bit_depth_luma_minus8** is not present in the film grain characteristics SEI message, the value of **fg_bit_depth_luma_minus8** is inferred to be equal to $\text{BitDepth}_Y - 8$.

The value of $\text{fgBitDepth}[0]$ is derived as follows:

$$\text{fgBitDepth}[0] = \text{fg_bit_depth_luma_minus8} + 8 \quad (19)$$

fg_bit_depth_chroma_minus8 plus 8 specifies the bit depth used for the Cb and Cr components of the film grain characteristics specified in the SEI message. When **fg_bit_depth_chroma_minus8** is not present in the film grain characteristics SEI message, the value of **fg_bit_depth_chroma_minus8** is inferred to be equal to $\text{BitDepth}_C - 8$.

The value of $\text{fgBitDepth}[c]$ for $c = 1$ and 2 is derived as follows:

$$\text{fgBitDepth}[c] = \text{fg_bit_depth_chroma_minus8} + 8, \text{ with } c = 1, 2 \quad (20)$$

fg_full_range_flag has the same semantics as specified in clause 7.3 for the **vui_full_range_flag** syntax element, except as follows:

- **fg_full_range_flag** specifies the video full range flag of the film grain characteristics specified in the SEI message, rather than the video full range flag used for the CLVS.
- When **fg_full_range_flag** is not present in the film grain characteristics SEI message, the value of **fg_full_range_flag** is inferred to be equal to **vui_full_range_flag**.

fg_colour_primaries has the same semantics as specified in clause 7.3 for the **vui_colour_primaries** syntax element, except as follows:

- **fg_colour_primaries** specifies the colour primaries of the film grain characteristics specified in the SEI message, rather than the colour primaries used for the CLVS.
- When **fg_colour_primaries** is not present in the film grain characteristics SEI message, the value of **fg_colour_primaries** is inferred to be equal to **vui_colour_primaries**.

fg_transfer_characteristics has the same semantics as specified in clause 7.3 for the **vui_transfer_characteristics** syntax element, except as follows:

- **fg_transfer_characteristics** specifies the transfer characteristics of the film grain characteristics specified in the SEI message, rather than the transfer characteristics used for the CLVS.
- When **fg_transfer_characteristics** is not present in the film grain characteristics SEI message, the value of **fg_transfer_characteristics** is inferred to be equal to **vui_transfer_characteristics**.

fg_matrix_coeffs has the same semantics as specified in clause 7.3 for the **vui_matrix_coeffs** syntax element, except as follows:

- **fg_matrix_coeffs** specifies the matrix coefficients of the film grain characteristics specified in the SEI message, rather than the matrix coefficients used for the CLVS.
- When **fg_matrix_coeffs** is not present in the film grain characteristics SEI message, the value of **fg_matrix_coeffs** is inferred to be equal to **vui_matrix_coeffs**.
- The values allowed for **fg_matrix_coeffs** are not constrained by the chroma format of the decoded video pictures that is indicated by the value of **ChromaFormatIdc** for the semantics of the VUI parameters.

fg_matrix_coeffs shall not be equal to 0, 16, or 17 unless **fg_bit_depth_luma_minus8** is equal to **fg_bit_depth_chroma_minus8**.

`fg_matrix_coeffs` shall not be equal to 8 unless one of the following conditions is true:

- `fg_bit_depth_luma_minus8` is equal to `fg_bit_depth_chroma_minus8`.
- `fg_bit_depth_chroma_minus8` is equal to `fg_bit_depth_luma_minus8` + 1.

fg_blending_mode_id identifies the blending mode used to blend the simulated film grain with the input images as specified in Table 6. `fg_blending_mode_id` shall be in the range of 0 to 1, inclusive. The values of 2 and 3 for `fg_blending_mode_id` are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders shall ignore film grain characteristics SEI messages with `fg_blending_mode_id` equal to 2 or 3.

Table 6 — fg_blending_mode_id values

Value	Description
0	Additive
1	Multiplicative

Depending on the value of `fg_blending_mode_id`, the blending mode is specified as follows:

- If `fg_blending_mode_id` is equal to 0, the blending mode is additive as specified by:

$$I_{\text{grain}}[c][x][y] = \text{Clip3}(0, (1 \ll \text{fgBitDepth}[c]) - 1, \hat{I}[c][x][y] + G[c][x][y]) \quad (21)$$

- Otherwise (`fg_blending_mode_id` is equal to 1), the blending mode is multiplicative as specified by:

$$I_{\text{grain}}[c][x][y] = \text{Clip3}(0, (1 \ll \text{fgBitDepth}[c]) - 1, \hat{I}[c][x][y] + \text{Round}((\hat{I}[c][x][y] * G[c][x][y]) \div ((1 \ll \text{fgBitDepth}[c]) - 1))) \quad (22)$$

where $\hat{I}[c][x][y]$ represents the sample value at coordinates x, y of the colour component c of the input image $\hat{I}$, $G[c][x][y]$ is the simulated film grain value at the same position and colour component, and `fgBitDepth`[c] is the number of bits used for each sample in a fixed-length unsigned binary representation of the arrays $I_{\text{grain}}[c][x][y]$, $\hat{I}[c][x][y]$, and $G[c][x][y]$, where $c = 0..2$, $x = 0..\text{fgPicWidth} - 1$, and $y = 0..\text{fgPicHeight} - 1$.

fg_log2_scale_factor specifies a scale factor used in the film grain characterization equations.

fg_comp_model_present_flag[c] equal to 0 indicates that film grain is not modelled on the c -th colour component, where c equal to 0 refers to the luma component, c equal to 1 refers to the Cb component, and c equal to 2 refers to the Cr component. `fg_comp_model_present_flag`[c] equal to 1 indicates that syntax elements specifying modelling of film grain on colour component c are present in the SEI message.

When `fg_separate_colour_description_present_flag` is equal to 0 and `ChromaFormatIdc` is equal to 0, the value of `fg_comp_model_present_flag`[1] and `fg_comp_model_present_flag`[2] shall be equal to 0.

fg_num_intensity_intervals_minus1[c] plus 1 specifies the number of intensity intervals for which a specific set of model values has been estimated.

NOTE 6 The intensity intervals could overlap in order to simulate multi-generational film grain.

fg_num_model_values_minus1[c] plus 1 specifies the number of model values present for each intensity interval in which the film grain has been modelled. The value of `fg_num_model_values_minus1`[c] shall be in the range of 0 to 5, inclusive.

fg_intensity_interval_lower_bound[c][i] specifies the lower bound of the i -th intensity interval for which the set of model values applies.

fg_intensity_interval_upper_bound[c][i] specifies the upper bound of the i -th intensity interval for which the set of model values applies.

The variable `intensityIntervalIdx[ c ][ x ][ y ][ j ]` represents the j -th index to the list of intensity intervals selected for the sample value $\hat{I}[c][x][y]$ for $c = 0..2$, $x = 0..fgPicWidth - 1$, $y = 0..fgPicHeight - 1$, and $j = 0..numApplicableIntensityIntervals[c][x][y] - 1$, where `numApplicableIntensityIntervals[ c ][ x ][ y ]` is derived below.

Depending on the value of `fg_model_id`, the selection of the one or more intensity intervals for the sample value $\hat{I}[c][x][y]$ is specified as follows:

- The variable `numApplicableIntensityIntervals[ c ][ x ][ y ]` is initially set equal to 0.
- If `fg_model_id` is equal to 0, the following applies:
 - The top-left sample location (xB, yB) of the current 8x8 block b that contains the sample value $\hat{I}[c][x][y]$ is derived as $(xB, yB) = (x / 8, y / 8)$.
 - The average value b_{avg} of the current 8x8 block b is derived as follows:

```
sum8x8 = 0
for( i = 0; i < 8; i++ )
    for( j = 0; j < 8; j++ )
        sum8x8 +=  $\hat{I}[ c ][ xB * 8 + i ][ yB * 8 + j ]$ 
b_avg = Clip3( 0, 255, ( sum8x8 + ( 1 << ( fgBitDepth[ c ] - 3 ) ) ) >> ( fgBitDepth[ c ] - 2 ) )
```

(23)

- The value of `intensityIntervalIdx[ c ][ x ][ y ][ j ]` is derived as follows:

```
for( i = 0, j = 0; i <= fg_num_intensity_intervals_minus1[ c ]; i++ )
    if( b_avg >= fg_intensity_interval_lower_bound[ c ][ i ] &&
        b_avg <= fg_intensity_interval_upper_bound[ c ][ i ] ) {
        intensityIntervalIdx[ c ][ x ][ y ][ j ] = i
        j++
    }
numApplicableIntensityIntervals[ c ][ x ][ y ] = j
```

(24)

- Otherwise (`fg_model_id` is equal to 1), the value of `intensityIntervalIdx[ c ][ x ][ y ][ j ]` is derived as follows:

```
I8[ c ][ x ][ y ] = ( fgBitDepth[ c ] == 8 ) ? (  $\hat{I}[ c ][ x ][ y ]$  :
    Clip3( 0, 255, (  $\hat{I}[ c ][ x ][ y ]$  + ( 1 << ( fgBitDepth[ c ] - 9 ) ) ) >> ( fgBitDepth[ c ] - 8 ) )
for( i = 0, j = 0; i <= fg_num_intensity_intervals_minus1[ c ]; i++ )
    if( I8[ c ][ x ][ y ] >= fg_intensity_interval_lower_bound[ c ][ i ] &&
        I8[ c ][ x ][ y ] <= fg_intensity_interval_upper_bound[ c ][ i ] ) {
        intensityIntervalIdx[ c ][ x ][ y ][ j ] = i
        j++
    }
numApplicableIntensityIntervals[ c ][ x ][ y ] = j
```

(25)

Samples that do not fall into any of the defined intervals (i.e., those samples for which the value of `numApplicableIntensityIntervals[ c ][ x ][ y ]` is equal to 0) are not modified by the grain generation function. Samples that fall into more than one interval (i.e., those samples for which the value of `numApplicableIntensityIntervals[ c ][ x ][ y ]` is greater than 1) will originate multi-generation grain. Multi-generation grain results from adding the grain computed independently for each of the applicable intensity intervals.

In the pseudocode steps in the remainder of this clause, the variable s_j in each instance of the list `fg_comp_model_value[ c ][ sj ]` is the value of `intensityIntervalIdx[ c ][ x ][ y ][ j ]` derived for the sample value $\hat{I}[c][x][y]$.

fg_comp_model_value[c][i][j] specifies the j -th model value for the colour component c and the i -th intensity interval. The set of model values has different meaning depending on the value of `fg_model_id`.

The value of `fg_comp_model_value[ c ][ i ][ j ]` is constrained as follows, and could be additionally constrained as specified elsewhere in this clause:

- If `fg_model_id` is equal to 0, `fg_comp_model_value[ c ][ i ][ j ]` shall be in the range of 0 to $2^{\text{fgBitDepth}[c]} - 1$, inclusive.
- Otherwise (`fg_model_id` is equal to 1), `fg_comp_model_value[ c ][ i ][ j ]` shall be in the range of $-2^{\text{fgBitDepth}[c] - 1}$ to $2^{\text{fgBitDepth}[c] - 1} - 1$, inclusive.

Depending on the value of `fg_model_id`, the synthesis of the film grain is modelled as follows:

- If `fg_model_id` is equal to 0, a frequency filtering model enables simulating the original film grain for $c = 0..2$, $x = 0..\text{fgPicWidth} - 1$, and $y = 0..\text{fgPicHeight} - 1$ as specified by:

$$G[c][x][y] = (\text{fg_comp_model_value}[c][s_j][0] * Q[c][x][y] + \text{fg_comp_model_value}[c][s_j][5] * G[c - 1][x][y]) \gg \text{fg_log2_scale_factor} \quad (26)$$

where $Q[c]$ is a two-dimensional random process generated by filtering 16×16 blocks `gaussRv` with random-valued elements `gaussRvij` generated with a normalized Gaussian distribution (independent and identically distributed Gaussian random variable samples with zero mean and unity variance) and where the value of an element $G[c - 1][x][y]$ used in the right-hand side of the formula is inferred to be equal to 0 when $c - 1$ is less than 0.

NOTE 7 A normalized Gaussian random variable can be generated from two independent, uniformly distributed random values over the interval from 0 to 1 (and not equal to 0), denoted as uRv_0 and uRv_1 , using the Box-Muller transformation specified by:

$$\text{gaussRv}_{ij} = \text{Sqrt}(-2 * \text{Ln}(uRv_0)) * \text{Cos}(2 * \pi * uRv_1) \quad (27)$$

where π is Archimedes' constant 3.141 592 653 589 793....

The band-pass filtering of blocks `gaussRv` can be performed in the discrete cosine transform (DCT) domain as follows:

```
for( y = 0; y < 16; y++ )
  for( x = 0; x < 16; x++ )
    if( ( x < fg_comp_model_value[ c ][ sj ][ 3 ] && y < fg_comp_model_value[ c ][ sj ][ 4 ] ) ||
        x > fg_comp_model_value[ c ][ sj ][ 1 ] || y > fg_comp_model_value[ c ][ sj ][ 2 ] )
      gaussRv[ x ][ y ] = 0
    filteredRv = IDCT16x16( gaussRv )
```

where `IDCT16x16( z )` refers to a unitary inverse discrete cosine transformation (IDCT) operating on a 16×16 matrix argument z as specified by:

$$\text{IDCT}_{16 \times 16}(z) = r * z * r^T \quad (29)$$

where the superscript T indicates a matrix transposition and r is the 16x16 matrix with elements r_{ij} specified by:

$$r_{i,j} = \frac{((i=0)?1:\text{Sqrt}(2))}{4} * \text{Cos}\left(\frac{i*(2*j+1)*\pi}{32}\right) \quad (30)$$

where π is Archimedes' constant 3.141 592 653 589 793....

$Q[c]$ is formed by the frequency-filtered blocks filteredRv.

NOTE 8 Coded model values are based on blocks of size 16x16, but a decoder implementation could use other block sizes. For example, decoders implementing the IDCT on 8x8 blocks could down-convert by a factor of two the set of coded model values $\text{fg_comp_model_value}[c][s_j][i]$ for i equal to 1..4.

NOTE 9 To reduce the degree of visible blocks that result from mosaicking the frequency-filtered blocks filteredRv, decoders could apply a low-pass filter to the boundaries between frequency-filtered blocks.

- Otherwise (fg_model_id is equal to 1), an auto-regression model enables simulating the original film grain for $c = 0..2$, $x = 0..\text{fgPicWidth} - 1$, and $y = 0..\text{fgPicHeight} - 1$ as specified by:

$$\begin{aligned} G[c][x][y] = & (\text{fg_comp_model_value}[c][s_j][0] * n[c][x][y] + \\ & \text{fg_comp_model_value}[c][s_j][1] * (G[c][x-1][y] + \\ & ((\text{fg_comp_model_value}[c][s_j][4] * G[c][x][y-1]) >> \\ & \text{fg_log2_scale_factor})) + \text{fg_comp_model_value}[c][s_j][3] * \\ & ((\text{fg_comp_model_value}[c][s_j][4] * (G[c][x-1][y-1] + G[c][x+1][y-1])) >> \\ & \text{fg_log2_scale_factor})) + \\ & \text{fg_comp_model_value}[c][s_j][5] * (G[c][x-2][y] + \\ & ((\text{fg_comp_model_value}[c][s_j][4] * \text{fg_comp_model_value}[c][s_j][4] * G[c][x][y-2]) >> \\ & (2 * \text{fg_log2_scale_factor}))) + \\ & \text{fg_comp_model_value}[c][s_j][2] * G[c-1][x][y]) >> \text{fg_log2_scale_factor} \end{aligned} \quad (31)$$

where $n[c][x][y]$ is a random value with normalized Gaussian distribution (independent and identically distributed Gaussian random variable samples with zero mean and unity variance for each value of c, x, and y) and where the value of an element $G[c][x][y]$ used in the right-hand side of the formula is inferred to be equal to 0 when any of the following conditions are true:

- c is less than 0,
- x is less than 0,
- y is less than 0.

$\text{fg_comp_model_value}[c][i][0]$ provides the first model value for the model as specified by fg_model_id . $\text{fg_comp_model_value}[c][i][0]$ corresponds to the standard deviation of the Gaussian noise term in the generation functions specified in pseudocode steps 26 through 31.

$\text{fg_comp_model_value}[c][i][1]$ provides the second model value for the model as specified by fg_model_id . When fg_model_id is equal to 0, $\text{fg_comp_model_value}[c][i][1]$ shall be greater than or equal to 0 and less than 16.

When not present in the film grain characteristics SEI message, $fg_comp_model_value[c][i][1]$ is inferred as follows:

- If fg_model_id is equal to 0, $fg_comp_model_value[c][i][1]$ is inferred to be equal to 8.
- Otherwise (fg_model_id is equal to 1), $fg_comp_model_value[c][i][1]$ is inferred to be equal to 0.

$fg_comp_model_value[c][i][1]$ is interpreted as follows:

- If fg_model_id is equal to 0, $fg_comp_model_value[c][i][1]$ indicates the horizontal high cut frequency to be used to filter the DCT of a block of 16x16 random values.
- Otherwise (fg_model_id is equal to 1), $fg_comp_model_value[c][i][1]$ indicates the first order spatial correlation for neighbouring samples $(x - 1, y)$ and $(x, y - 1)$.

$fg_comp_model_value[c][i][2]$ provides the third model value for the model as specified by fg_model_id . When fg_model_id is equal to 0, $fg_comp_model_value[c][i][2]$ shall be greater than or equal to 0 and less than 16.

When not present in the film grain characteristics SEI message, $fg_comp_model_value[c][i][2]$ is inferred as follows:

- If fg_model_id is equal to 0, $fg_comp_model_value[c][i][2]$ is inferred to be equal to $fg_comp_model_value[c][i][1]$.
- Otherwise (fg_model_id is equal to 1), $fg_comp_model_value[c][i][2]$ is inferred to be equal to 0.

$fg_comp_model_value[c][i][2]$ is interpreted as follows:

- If fg_model_id is equal to 0, $fg_comp_model_value[c][i][2]$ indicates the vertical high cut frequency to be used to filter the DCT of a block of 16x16 random values.
- Otherwise (fg_model_id is equal to 1), $fg_comp_model_value[c][i][2]$ indicates the colour correlation between consecutive colour components.

$fg_comp_model_value[c][i][3]$ provides the fourth model value for the model as specified by fg_model_id . When fg_model_id is equal to 0, $fg_comp_model_value[c][i][3]$ shall be greater than or equal to 0 and less than or equal to $fg_comp_model_value[c][i][1]$.

When not present in the film grain characteristics SEI message, $fg_comp_model_value[c][i][3]$ is inferred to be equal to 0.

$fg_comp_model_value[c][i][3]$ is interpreted as follows:

- If fg_model_id is equal to 0, $fg_comp_model_value[c][i][3]$ indicates the horizontal low cut frequency to be used to filter the DCT of a block of 16x16 random values.
- Otherwise (fg_model_id is equal to 1), $fg_comp_model_value[c][i][3]$ indicates the first order spatial correlation for neighbouring samples $(x - 1, y - 1)$ and $(x + 1, y - 1)$.

$fg_comp_model_value[c][i][4]$ provides the fifth model value for the model as specified by fg_model_id . When fg_model_id is equal to 0, $fg_comp_model_value[c][i][4]$ shall be greater than or equal to 0 and less than or equal to $fg_comp_model_value[c][i][2]$.

When not present in the film grain characteristics SEI message, $fg_comp_model_value[c][i][4]$ is inferred to be equal to fg_model_id .

$fg_comp_model_value[c][i][4]$ is interpreted as follows:

- If fg_model_id is equal to 0, $fg_comp_model_value[c][i][4]$ indicates the vertical low cut frequency to be used to filter the DCT of a block of 16x16 random values.
- Otherwise (fg_model_id is equal to 1), $fg_comp_model_value[c][i][4]$ indicates the aspect ratio of the modelled grain.

`fg_comp_model_value[ c ][ i ][ 5 ]` provides the sixth model value for the model as specified by `fg_model_id`.

When not present in the film grain characteristics SEI message, `fg_comp_model_value[ c ][ i ][ 5 ]` is inferred to be equal to 0.

`fg_comp_model_value[ c ][ i ][ 5 ]` is interpreted as follows:

- If `fg_model_id` is equal to 0, `fg_comp_model_value[ c ][ i ][ 5 ]` indicates the colour correlation between consecutive colour components.
- Otherwise (`fg_model_id` is equal to 1), `fg_comp_model_value[ c ][ i ][ 5 ]` indicates the second order spatial correlation for neighbouring samples $(x, y - 2)$ and $(x - 2, y)$.

fg_characteristics_persistence_flag specifies the persistence of the film grain characteristics SEI message for the current layer.

`fg_characteristics_persistence_flag` equal to 0 specifies that the film grain characteristics SEI message applies to the current decoded picture only.

`fg_characteristics_persistence_flag` equal to 1 specifies that the film grain characteristics SEI message applies to the current decoded picture and persists for all subsequent pictures of the current layer in output order until one or more of the following conditions are true:

- A new CLVS of the current layer begins.
- The bitstream ends.
- A picture in the current layer in an AU associated with a film grain characteristics SEI message is output that follows the current picture in output order.

fg_spatial_resolution_present_flag equal to 1 specifies that syntax elements defining the spatial resolution for the film grain synthesis are present in the SEI message. `fg_spatial_resolution_present_flag` equal to 0 indicates that no syntax elements for spatial resolution of the film grain synthesis are present in the SEI message. When not present in the film grain characteristics SEI message, `fg_spatial_resolution_present_flag` is inferred to be equal to 0.

fg_pic_width_in_luma_samples specifies the width of the picture in units of luma samples for which the film grain modelling information is indicated. When present, the value of `fg_pic_width_in_luma_samples` shall be in the range of 1 to 65 536, inclusive. The value of `fg_pic_width_in_luma_samples` equal to 0 is reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders shall ignore film grain characteristics SEI messages with `fg_pic_width_in_luma_samples` equal to 0.

fg_pic_height_in_luma_samples specifies the height of the picture in units of luma samples for which the film grain modelling information is indicated. When present, the value of `fg_pic_height_in_luma_samples` shall be in the range of 1 to 65 536, inclusive. The value of `fg_pic_height_in_luma_samples` equal to 0 is reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders shall ignore film grain characteristics SEI messages with `fg_pic_height_in_luma_samples` equal to 0.

NOTE 9 Film grain modelling information for more than one spatial resolution can be indicated in the same bitstream by including the film grain characteristics SEI messages of different spatial resolutions into different processing chains and including film grain characteristics SEI messages in processing order nesting SEI messages so that a film grain characteristics SEI message for a particular spatial resolution does not cancel the persistence of another film grain characteristics SEI message for another spatial resolution.

Modify subclause 8.18.2 as follows:

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

- The width and height of a cropped decoded picture in units of luma samples, denoted herein by `CroppedWidth` and `CroppedHeight`, respectively.
- A chroma format indicator, denoted herein by `ChromaFormatIdc`, as described in clause 7.3.

`ar_bounding_box_top[ ar_object_idx[ i ] ]`, `ar_bounding_box_left[ ar_object_idx[ i ] ]`, `ar_bounding_box_width[ ar_object_idx[ i ] ]`, and `ar_bounding_box_height[ ar_object_idx[ i ] ]` specify the coordinates of the top-left corner and the width and height, respectively, of the bounding box of the `ar_object_idx[ i ]`-th object in the cropped decoded picture, relative to the top-left luma sample of the cropped decoded picture.

The value of `ar_bounding_box_left[ ar_object_idx[ i ] ]` shall be in the range of 0 to `CroppedWidth / SubWidthC - 1`, inclusive.

The value of `ar_bounding_box_top[ ar_object_idx[ i ] ]` shall be in the range of 0 to `CroppedHeight / SubHeightC - 1`, inclusive.

The value of `ar_bounding_box_width[ ar_object_idx[ i ] ]` shall be in the range of 0 to `CroppedWidth / SubWidthC - ar_bounding_box_left[ ar_object_idx[ i ] ]`, inclusive.

The value of `ar_bounding_box_height[ ar_object_idx[ i ] ]` shall be in the range of 0 to `CroppedHeight / SubHeightC - ar_bounding_box_top[ ar_object_idx[ i ] ]`, inclusive.

The identified object rectangle contains the luma samples with horizontal picture coordinates from `SubWidthC * ar_bounding_box_left[ ar_object_idx[ i ] ]` to `SubWidthC * ( ar_bounding_box_left[ ar_object_idx[ i ] ] + ar_bounding_box_width[ ar_object_idx[ i ] ] ) - 1`, inclusive, relative to leftmost luma sample column of the cropped decoded picture and vertical picture coordinates from `SubHeightC * ar_bounding_box_top[ ar_object_idx[ i ] ]` to `SubHeightC * ( ar_bounding_box_top[ ar_object_idx[ i ] ] + ar_bounding_box_height[ ar_object_idx[ i ] ] ) - 1`, inclusive, relative to the topmost luma sample row of the cropped decoded picture.

Modify subclause 8.19.2 as follows:

`sdi_aux_id[ i ]` equal to 0 indicates that the *i*-th layer in the current CVS does not contain auxiliary pictures. `sdi_aux_id[ i ]` greater than 0 indicates the type of auxiliary pictures in the *i*-th layer in the current CVS as specified in Table 15. When `sdi_auxiliary_info_flag` is equal to 0, the value of `sdi_aux_id[ i ]` is inferred to be equal to 0.

Table 15 — Mapping of `sdi_aux_id[ i ]` to the type of auxiliary pictures

<code>sdi_aux_id[i]</code>	Name	Type of auxiliary pictures
1	AUX_ALPHA	Alpha plane
2	AUX_DEPTH	Depth picture
3	AUX_OBJECT_MASK	Object mask picture
4..127		Reserved
128..159		Unspecified
160..255		Reserved

NOTE 1 The interpretation of auxiliary pictures associated with `sdi_aux_id[ i ]` in the range of 128 to 159, inclusive, is specified through means other than the `sdi_aux_id[ i ]` value.

`sdi_aux_id[ i ]` shall be in the range of 0 to 3, inclusive, or 128 to 159, inclusive, for bitstreams conforming to this version of this document. Although the value of `sdi_aux_id[ i ]` shall be in the range of 0 to 3, inclusive, or 128 to 159, inclusive, in this version of this document, decoders shall also allow other values of `sdi_aux_id[ i ]` in the range of 0 to 255, inclusive.

If `sdi_aux_id[ i ]` is equal to 0, the *i*-th layer is referred to as a primary layer. Otherwise, the *i*-th layer is referred to as an auxiliary layer. When `sdi_aux_id[ i ]` is equal to 1, the *i*-th layer is also referred to as an alpha auxiliary layer. When `sdi_aux_id[ i ]` is equal to 2, the *i*-th layer is also referred to as a depth auxiliary layer. When `sdi_aux_id[ i ]` is equal to 3, the *i*-th layer is also referred to as an object mask auxiliary layer.

In subclauses 8.22.2 (Depth representation information SEI message semantics) and 8.23.2 (Alpha channel information SEI message semantics), replace each of the 10 instances of "nuh_layer_id" with "layer identification".

Modify subclause 8.26.2 (Colour transform information SEI message semantics) to be as follows:

The colour transform information (CTI) SEI message provides information to enable remapping of the reconstructed colour samples of the output pictures for purposes such as converting the output pictures to a representation that is more suitable for an alternative display. The colour transform model used in the CTI SEI message is composed of a first piece-wise linear function applied to the first colour component. Depending on the values of syntax elements `colour_transform_cross_component_flag`, `colour_transform_cross_comp_inferred_flag`, and `colour_transform_lut2_present_flag`, one or two additional piece-wise linear functions may be signalled for the second and third colour components.

When `ChromaFormatIdc` is equal to 0 (monochrome), the CTI SEI message shall not be present, although decoders shall also allow such messages to be present and shall ignore any such CTI SEI messages when present.

colour_transform_id contains an identifying number that may be used to identify the purpose of the CTI. The value of `colour_transform_id` may be used (in a manner not specified in this document) to indicate that the input to the remapping process is the output of some conversion process that is not specified in this document, such as a conversion of the picture to some alternative colour representation (e.g., conversion from a YCbCr colour representation to a GBR colour representation). When more than one CTI SEI message is present with the same value of `colour_transform_id`, the content of these CTI SEI messages shall be the same. When CTI SEI messages are present that have more than one value of `colour_transform_id`, this may indicate that the remapping processes indicated by the different values of `colour_transform_id` are alternatives that are provided for different purposes or that a cascading of remapping processes is to be applied in a sequential order (an order that is not specified in this document). The value of `colour_transform_id` shall be in the range of 0 to $2^{32} - 2$, inclusive.

Values of `colour_transform_id` from 0 to 255 and from 512 to $2^{31} - 1$ may be used as determined by the application. Values of `colour_transform_id` from 256 to 511, inclusive, and from 2^{31} to $2^{32} - 2$, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the CTI SEI messages containing a value of `colour_transform_id` in the range of 256 to 511, inclusive, or in the range of 2^{31} to $2^{32} - 2$, inclusive, and bitstreams conforming to this version of this document shall not contain `colour_transform_id` with such values.

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

colour_transform_cancel_flag equal to 1 indicates that the CTI SEI message cancels the persistence of any previous CTI SEI message in output order that applies to the current layer. `colour_transform_cancel_flag` equal to 0 indicates that CTI follows.

colour_transform_persistence_flag specifies the persistence of the CTI SEI message for the current layer.

`colour_transform_persistence_flag` equal to 0 specifies that the CTI SEI message applies to the current decoded picture only.

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

- A new CLVS of the current layer begins.
- The bitstream ends.

- A picture in the current layer in an AU associated with a CTI SEI message with the same value of colour_transform_id and applicable to the current layer is output that follows the current picture in output order.

colour_transform_video_signal_info_present_flag equal to 1 specifies that syntax elements colour_transform_full_range_flag, colour_transform primaries, colour_transform_transfer_function and colour_transform_matrix_coefficients are present. colour_transform_video_signal_info_present_flag equal to 0 specifies that syntax elements colour_transform_full_range_flag, colour_transform primaries, colour_transform_transfer_function and colour_transform_matrix_coefficients are not present.

colour_transform_full_range_flag has the same semantics as specified in clause 7.3 for the vui_full_range_flag syntax element, except that colour_transform_full_range_flag identifies the video full range flag of the remapped reconstructed picture, rather than the video full range flag used for the CLVS. When not present, the value of colour_transform_full_range_flag is inferred to be equal to the value of vui_full_range_flag.

colour_transform_primaries has the same semantics as specified in clause 7.3 for the vui_colour_primaries syntax element, except that colour_transform_primaries identifies the colour primaries of the remapped reconstructed picture, rather than the colour primaries used for the CLVS. When not present, the value of colour_transform_primaries is inferred to be equal to the value of vui_colour_primaries.

colour_transform_transfer_function has the same semantics as specified in clause 7.3 for the vui_transfer_characteristics syntax element, except that colour_transform_transfer_function identifies the transfer characteristics of the remapped reconstructed picture, rather than the transfer characteristics used for the CLVS. When not present, the value of colour_transform_transfer_function is inferred to be equal to the value of vui_transfer_characteristics.

colour_transform_matrix_coefficients has the same semantics as specified in clause 7.3 for the vui_matrix_coeffs syntax element, except that colour_transform_matrix_coefficients identifies the matrix coefficients of the remapped reconstructed picture, rather than the matrix coefficients used for the CLVS. When not present, the value of colour_transform_matrix_coefficients is inferred to be equal to the value of vui_matrix_coeffs.

colour_transform_bit_depth_minus8 plus 8 specifies the bit depth of the colour components of the associated pictures for purposes of interpretation of the CTI SEI message. When any CTI SEI message is present with the value of colour_transform_bit_depth plus 8 not equal to the bit depth of the decoded colour components, the SEI message refers to the hypothetical result of a conversion operation performed to convert the decoded colour component samples to the bit depth equal to colour_transform_input_bit_depth plus 8.

The value of colour_transform_bit_depth plus 8 shall be in the range of 8 to 16, inclusive. Values of colour_transform_bit_depth from in the range of 17 to 23, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the CTI SEI messages that contain a value of colour_transform_bit_depth in the range of 17 to 23, inclusive, and bitstreams conforming to this version of this document shall not contain colour_transform_bit_depth with such values.

bitDepth is set equal to (colour_transform_bit_depth + 8).

colour_transform_log2_number_of_points_per_lut_minus1 specifies the log2 of the number of pivot points in the piece-wise linear remapping functions minus 1.

log2numLutPoints is set equal to (colour_transform_log2_number_of_points_per_lut_minus1 + 1).

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

colourTransformSize is set equal to (numLutPoints + 1).

log2distX is set equal to (bitDepth – log2numLutPoints).

colour_transform_cross_component_flag equal to 1 indicates that the remapping of the second and third colour components is performed as cross-component remapping based on the first colour component. colour_transform_cross_component_flag equal to 0 indicates that intra-component remapping is applied to the second and third colour components.

maxIntraComp is set equal to (2 * (1 – colour_transform_cross_component_flag)).

colour_transform_cross_comp_inferred_flag equal to 1 indicates that the remapping piece-wise linear functions of the second and third colour components are inferred from the remapping piece-wise linear function of the first colour component. **colour_transform_cross_comp_inferred_flag** equal to 0 indicates that the remapping piece-wise linear functions of the second and third colour components are signalled. When not present, the value of **colour_transform_cross_comp_inferred_flag** is inferred to be equal to 0.

colour_transf_lut[c][i] specifies the piecewise linear remapping function of the colour component of index c. When **colour_transf_lut**[1][i] is present and **colour_transf_lut**[2][i] is not present, the value of **colour_transf_lut**[2][i] is inferred to be equal to **colour_transf_lut**[1][i]. The length of **colour_transf_lut**[c][i] is $2 + \text{bitDepth} - \log_2 \text{numLutPoints}$ bits.

colour_transform_lut2_present_flag equal to 1 specifies that **colour_transf_lut**[2][i] is present in the CTI SEI message. **colour_transform_lut2_present_flag** equal to 0 specifies that **colour_transf_lut**[2][i] is not present in the CTI SEI message. When not present, the value of **colour_transform_lut2_present_flag** is inferred to be equal to 0.

colour_transform_chroma_offset specifies the CTI chroma offset. When not present, **colour_transform_chroma_offset** is inferred to be equal to 0. The length of **colour_transform_chroma_offset** is $2 + \text{bitDepth} - \log_2 \text{numLutPoints}$ bits.

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

The array **pivotPointX** is derived as follows.

— For $j=0..(\text{numLutPoints} - 1)$, **pivotPointX**[j] is set equal to $(j \ll \log_2 \text{distX})$.

For $c=0..\text{maxIntraComp}$, the arrays **pivotPointY**[c] and **slope**[c] are derived as follows:

— **pivotPointY**[c][0] is set equal to **colour_transf_lut**[c][0]

— For $j=1..(\text{numLutPoints} - 1)$, **pivotPointY**[c][j] is derived as follows:

$$\text{pivotPointY}[c][j] = \text{pivotPointY}[c][j-1] + \text{colour_transf_lut}[c][j] \quad (68)$$

— For $j=0..(\text{numLutPoints} - 1)$, **slope**[c][j] is derived as follows:

$$\text{slope}[c][j] = ((\text{colour_transf_lut}[c][j+1] \ll 11) + (1 \ll (\log_2 \text{distX} - 1))) \gg \log_2 \text{distX} \quad (69)$$

When **colour_transform_cross_component_flag** is equal to 1, the arrays **ccPivotPointY**[c] and **ccSlope**[c] are derived as follows, for $c=1..2$:

— If **colour_transform_cross_comp_inferred_flag** is equal to 0, **ccPivotPointY**[c] is derived as follows:

— For $j=0..\text{numLutPoints}$, **ccPivotPointY**[c][j] is set equal to $(\text{colour_transf_lut}[c][j] \ll (11 - \log_2 \text{distX}))$.

— Otherwise (**colour_transform_cross_comp_inferred_flag** is equal to 1), **ccPivotPointY**[c] is derived as follows:

— For $j=0..(\text{numLutPoints} - 1)$, **tmpPivotPt**[j] is derived as follows:

— If **colour_transf_lut**[0][j + 1] is equal to 0, **tmpPivotPt**[j] is set equal to $(1 \ll 11)$.

— Otherwise, **tmpPivotPt**[j] is derived as follows:

$$\text{tmpPivotPt}[j] = (\text{colour_transf_lut}[0][j+1] + \text{colour_transform_chroma_offset}) \ll 11 - \log_2 \text{distX} \quad (70)$$

— The array `ccPivotPointY[ c ]` is derived as follows:

— For $j=1..(\text{numLutPoints} - 1)$, `ccPivotPointY[ c ][ j ]` is derived as follows:

$$\text{ccPivotPointY}[c][j] = (\text{tmpPivotPt}[j] + \text{tmpPivotPt}[j - 1] + 1) / 2 \quad (71)$$

— `ccPivotPointY[ c ][ 0 ]` is set equal to `tmpPivotPt[ 0 ]`.

— `ccPivotPointY[ c ][ numLutPoints ]` is set equal to `tmpPivotPt[ numLutPoints - 1 ]`.

— For $j=0..(\text{numLutPoints} - 1)$, the value of `ccSlope[ c ][ j ]` is set equal to $(\text{ccPivotPointY}[c][j+1] - \text{ccPivotPointY}[c][j])$.

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

— for $i=0..\text{picWidth}[c] - 1, j=0..\text{picHeight}[c] - 1$, the following applies:

$$\text{idx} = \text{rec}[c][i][j] \gg \log2\text{distX}$$

$$\text{map}[c][i][j] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \text{pivotPointY}[c][\text{idx}] + \quad (72)$$

$$((\text{slope}[c][\text{idx}] * (\text{rec}[i][j] - \text{pivotPointX}[\text{idx}]) + (1 \ll 10)) \gg 11))$$

When `colour_transform_cross_component_flag` is equal to 1, for $c=1..2$, the cross-component remapping process of the input samples picture `rec[ c ]` into the remapped samples picture `map[ c ]` is performed as follows:

— `offset` is set equal to $(1 \ll (\text{bitDepth} - 1))$.

— `subWc` and `subHc` are set equal to $(\text{picWidth}[0] / \text{picWidth}[c])$ and $(\text{picHeight}[0] / \text{picHeight}[c])$, respectively.

— For $i=0..\text{picWidth}[c] - 1, j=0..\text{picHeight}[c] - 1$, the following applies:

$$\text{coloc} = \text{rec}[0][i * \text{SubWc}][j * \text{SubHc}]$$

$$\text{idx} = \text{coloc} \gg \log2\text{distX} \quad (73)$$

$$\text{scale} = \text{ccPivotPointY}[c][\text{idx}] + ((\text{ccSlope}[c][\text{idx}] * (\text{coloc} - \text{pivotPointX}[\text{idx}])) \gg \log2\text{distX})$$

$$\text{map}[c][i][j] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1,$$

$$((\text{offset} \ll 11) + \text{scale} * (\text{rec}[c][i][j] - \text{offset}) + (1 \ll 10)) \gg 11)$$

Modify subclause 8.27.1 as follows:

shutter_interval_info(payloadSize) {	Descriptor
sii_time_scale	u(32)
sii_fixed_shutter_interval_within_clvs_flag	u(1)
if(sii_fixed_shutter_interval_within_clvs_flag)	
sii_num_units_in_shutter_interval	u(32)
else {	
sii_max_sub_layers_minus1	u(3)
for(i = 0; i <= sii_max_sub_layers_minus1; i++)	

shutter_interval_info(payloadSize) {	Descriptor
sii_sub_layer_num_units_in_shutter_interval[i]	u(32)
}	
}	

Remove subclause 8.28.1 (Note: The text was moved and merged, with updates, to subclause 8.30.3).

Renumber clause 8.28.2 to be clause 8.28.1 and modify it to be as follows, while retaining Figure 12 without modification:

8.28.1 Neural-network post-filter characteristics SEI message

8.28.1.1 Neural-network post-filter characteristics SEI message syntax

nn_post_filter_characteristics(payloadSize) {	Descriptor
nnpfc_purpose	u(16)
nnpfc_id	ue(v)
nnpfc_base_flag	u(1)
nnpfc_mode_idc	ue(v)
if(nnpfc_mode_idc == 1) {	
while(!byte_aligned())	
nnpfc_alignment_zero_bit_a	f(1)
nnpfc_tag_uri	st(v)
nnpfc_uri	st(v)
}	
nnpfc_property_present_flag	u(1)
if(nnpfc_property_present_flag) {	
/* input and output formatting */	
nnpfc_num_input_pics_minus1	ue(v)
if(nnpfc_num_input_pics_minus1 > 0) {	
for(i = 0; i <= nnpfc_num_input_pics_minus1; i++)	
nnpfc_input_pic_filtering_flag[i]	u(1)
nnpfc_absent_input_pic_zero_flag	u(1)
}	
if(ChromaUpsamplingFlag)	
nnpfc_out_sub_c_flag	u(1)
if(ColourizationFlag)	
nnpfc_out_colour_format_idc	u(2)
if(ResolutionResamplingFlag) {	
nnpfc_pic_width_num_minus1	ue(v)
nnpfc_pic_width_denom_minus1	ue(v)
nnpfc_pic_height_num_minus1	ue(v)
nnpfc_pic_height_denom_minus1	ue(v)
}	
if(PictureRateUpsamplingFlag)	
for(i = 0; i < nnpfc_num_input_pics_minus1; i++)	
nnpfc_interpolated_pics[i]	ue(v)
if(TemporalExtrapolationFlag)	

nn_post_filter_characteristics(payloadSize) {	Descriptor
nnpfc_extrapolated_pics_minus1	ue(v)
if(SpatialExtrapolationFlag) {	
nnpfc_spatial_extrapolation_left_offset	se(v)
nnpfc_spatial_extrapolation_right_offset	se(v)
nnpfc_spatial_extrapolation_top_offset	se(v)
nnpfc_spatial_extrapolation_bottom_offset	se(v)
}	
nnpfc_component_last_flag	u(1)
nnpfc_inp_format_idc	ue(v)
nnpfc_auxiliary_inp_idc	ue(v)
if((nnpfc_auxiliary_inp_idc & 2) > 0) {	
nnpfc_inband_prompt_flag	u(1)
if(nnpfc_inband_prompt_flag) {	
while(!byte_aligned())	
nnpfc_alignment_zero_bit_c	f(1)
nnpfc_prompt	st(v)
}	
}	
if((nnpfc_auxiliary_inp_idc & 4) > 0) {	
nnpfc_inband_seed_flag	u(1)
if(nnpfc_inband_seed_flag)	
nnpfc_seed	u(16)
}	
nnpfc_inp_order_idc	ue(v)
if(nnpfc_inp_format_idc == 1) {	
if(nnpfc_inp_order_idc != 1)	
nnpfc_inp_tensor_luma_bitdepth_minus8	ue(v)
if(nnpfc_inp_order_idc > 0)	
nnpfc_inp_tensor_chroma_bitdepth_minus8	ue(v)
}	
nnpfc_out_format_idc	ue(v)
nnpfc_out_order_idc	ue(v)
if(nnpfc_out_format_idc == 1) {	
if(nnpfc_out_order_idc != 1)	
nnpfc_out_tensor_luma_bitdepth_minus8	ue(v)
if(nnpfc_out_order_idc != 0)	
nnpfc_out_tensor_chroma_bitdepth_minus8	ue(v)
}	
nnpfc_separate_colour_description_present_flag	u(1)
if(nnpfc_separate_colour_description_present_flag) {	
nnpfc_colour_primaries	u(8)
nnpfc_transfer_characteristics	u(8)
if(nnpfc_out_format_idc == 1) {	
nnpfc_matrix_coeffs	u(8)
nnpfc_full_range_flag	u(1)

nn_post_filter_characteristics(payloadSize) {	Descriptor
}	
}	
if(nnpfc_out_order_idc > 0)	
nnpfc_chroma_loc_info_present_flag	u(1)
if(nnpfc_chroma_loc_info_present_flag)	
nnpfc_chroma_sample_loc_type_frame	ue(v)
if(!SpatialExtrapolationFlag) {	
nnpfc_overlap	ue(v)
nnpfc_constant_patch_size_flag	u(1)
}	
if(nnpfc_constant_patch_size_flag) {	
nnpfc_patch_width_minus1	ue(v)
nnpfc_patch_height_minus1	ue(v)
} else {	
nnpfc_extended_patch_width_cd_delta_minus1	ue(v)
nnpfc_extended_patch_height_cd_delta_minus1	ue(v)
}	
nnpfc_padding_type	ue(v)
if(nnpfc_padding_type == 4) {	
if(nnpfc_inp_order_idc != 1)	
nnpfc_luma_padding_val	ue(v)
if(nnpfc_inp_order_idc != 0) {	
nnpfc_cb_padding_val	ue(v)
nnpfc_cr_padding_val	ue(v)
}	
}	
nnpfc_complexity_info_present_flag	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_num_metadata_extension_bits	ue(v)
if(nnpfc_num_metadata_extension_bits > 0) {	
if(nnpfc_purpose == 0) {	
nnpfc_application_purpose_tag_uri_present_flag	u(1)
if(nnpfc_application_purpose_tag_uri_present_flag) {	
while(!byte_aligned())	
nnpfc_metadata_alignment_zero_bit	f(1)
nnpfc_application_purpose_tag_uri	st(v)
}	
}	

nn_post_filter_characteristics(payloadSize) {	Descriptor
if(SpatialExtrapolationFlag ResolutionResamplingFlag)	
nnpfc_scan_type_idc	u(2)
nnpfc_for_human_viewing_idc	u(2)
nnpfc_for_machine_analysis_idc	u(2)
nnpfc_reserved_metadata_extension	u(v)
}	
}	
/* ISO/IEC 15938-17 bitstream */	
if(nnpfc_mode_idc == 0) {	
while(!byte_aligned())	
nnpfc_alignment_zero_bit_b	f(1)
for(i = 0; more_data_in_payload(); i++)	
nnpfc_payload_byte[i]	b(8)
}	
}	

8.28.1.2 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 `CandInPicList` that contains a list of pictures in output order from which the input pictures for the NNPF are selected or padded.
- Input picture width and height in units of luma samples, denoted herein by `CroppedWidth` and `CroppedHeight`, respectively, of the input pictures.
- 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.
- Chroma format indicator `ChromaFormatIdc`, as described in clause 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.

Let `currCdoPic` be a cropped decoded output picture for which the NNPF defined by this NNPFC SEI message is activated by an NNPFA SEI message. Let `currPic` be a picture in `CandInPicList` that is a corresponding picture of `currCdoPic` or is an associated inserted picture of `currCdoPic`.

Input picture with index 0 is the picture `currPic`. 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.

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

NOTE 1 More than one NNPFC SEI message can be present for the same picture. When more than one NNPFC SEI message with different values of `nnpfc_id` is present or activated for the same picture, they can have the same value or different values of `nnpfc_purpose` and the same value or different values of `nnpfc_mode_idc`.

nnpfc_purpose indicates the purpose of the NNPFC as specified in Table 20, where $(\text{nnpfc_purpose} \& \text{bitMask})$ not equal to 0 indicates that the NNPFC has the purpose associated with the bitMask value in Table 20. When **nnpfc_purpose** is greater than 0 and $(\text{nnpfc_purpose} \& \text{bitMask})$ is equal to 0, the purpose associated with the bitMask value is not applicable to the NNPFC. When **nnpfc_purpose** is equal to 0, the NNPFC may be used as determined by the application and as specified by the **nnpfc_application_purpose_tag_uri**.

All NNPFC SEI messages with a particular value of **nnpfc_id** within a CLVS shall have the same value of **nnpfc_purpose**.

The value of **nnpfc_purpose** shall be in the range of 0 to 255, inclusive, in bitstreams conforming to this version of this document. Values of 256 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 version of this document. Decoders conforming to this version of this document shall ignore NNPFC SEI messages with **nnpfc_purpose** in the range of 256 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
0x40	Temporal extrapolation (i.e., generating one or more future pictures)
0x80	Spatial extrapolation (i.e., generating content outside of the spatial area of the input pictures), possibly also with removal (i.e. remove partial content from the input pictures)

The variables **ChromaUpsamplingFlag**, **ResolutionResamplingFlag**, **PictureRateUpsamplingFlag**, **BitDepthUpsamplingFlag**, **ColourizationFlag**, **TemporalExtrapolationFlag**, and **SpatialExtrapolationFlag**, specifying whether **nnpfc_purpose** indicates the purpose of the NNPFC to include chroma upsampling, resolution resampling, picture rate upsampling, bit depth upsampling, colourization, temporal extrapolation, and spatial extrapolation, respectively, are derived as follows:

$$\text{ChromaUpsamplingFlag} = ((\text{nnpfc_purpose} \& 0x02) > 0) ? 1 : 0$$

$$\text{ResolutionResamplingFlag} = ((\text{nnpfc_purpose} \& 0x04) > 0) ? 1 : 0$$

$$\text{PictureRateUpsamplingFlag} = ((\text{nnpfc_purpose} \& 0x08) > 0) ? 1 : 0 \quad (75)$$

$$\text{BitDepthUpsamplingFlag} = ((\text{nnpfc_purpose} \& 0x10) > 0) ? 1 : 0$$

$$\text{ColourizationFlag} = ((\text{nnpfc_purpose} \& 0x20) > 0) ? 1 : 0$$

$$\text{TemporalExtrapolationFlag} = ((\text{nnpfc_purpose} \& 0x40) > 0) ? 1 : 0$$

$$\text{SpatialExtrapolationFlag} = ((\text{nnpfc_purpose} \& 0x80) > 0) ? 1 : 0$$

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 or any one of a set of values including that value.

When **ChromaFormatIdc** is equal to 3, **ChromaUpsamplingFlag** shall be equal to 0.

When **ChromaUpsamplingFlag** is equal to 1, **ColourizationFlag** shall be equal to 0.

When **PictureRateUpsamplingFlag** or **TemporalExtrapolationFlag** 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`.

When `TemporalExtrapolationFlag` is equal to 1, the extrapolated pictures generated by the NNPf follow all input pictures of the NNPf in output order. When `TemporalExtrapolationFlag` is equal to 1 and there is a picture `picA` in `CandInPicList` that follows `currPic` in output order, the extrapolated pictures generated by the NNPf precede `picA` in output order.

`nnpfc_id` contains an identifying number that may be used to identify an NNPf. The value of `nnpfc_id` shall be in the range of 0 to $2^{32} - 2$, inclusive. Values of `nnpfc_id` from 256 to 511, inclusive, and from 2^{31} to $2^{32} - 2$, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders conforming to this version of this document encountering an NNPf SEI message with `nnpfc_id` in the range of 256 to 511, inclusive, or in the range of 2^{31} to $2^{32} - 2$, inclusive, shall ignore the SEI message.

When an NNPf SEI message is the first NNPf SEI message, in decoding order, that has a particular `nnpfc_id` value within the current CLVS, the following applies:

- This SEI message specifies a base NNPf.
- This SEI message pertains to the current decoded picture and all subsequent decoded pictures of the current layer, in output order, until the end of the current CLVS.

`nnpfc_base_flag` equal to 1 specifies that the SEI message specifies the base NNPf. `nnpfc_base_flag` equal to 0 specifies that the SEI message specifies an update relative to the base NNPf.

The following constraints apply to the value of `nnpfc_base_flag`:

- When an NNPf SEI message is the first NNPf SEI message, in decoding order, that has a particular `nnpfc_id` value within the current CLVS, the value of `nnpfc_base_flag` shall be equal to 1.
- All NNPf SEI messages in a CLVS that have a particular `nnpfc_id` value and `nnpfc_base_flag` equal to 1 shall have identical SEI payload content.

When `nnpfc_base_flag` is equal to 0, the following applies:

- This SEI message defines an update relative to the preceding base NNPf in decoding order with the same `nnpfc_id` value. Updates are not cumulative but rather each update is applied on the base NNPf, which is the NNPf specified by the first NNPf SEI message, in decoding order, that has a particular `nnpfc_id` value within the current CLVS. The NNPf defined by this SEI message is obtained by applying the update defined by this SEI message relative to the base NNPf with the same `nnpfc_id` value.
- This SEI message pertains to the current decoded picture and all subsequent decoded pictures of the current layer, in output order, until the end of the current CLVS or up to but excluding the decoded picture that follows the current decoded picture in output order within the current CLVS and is associated with a subsequent NNPf SEI message, in decoding order, having `nnpfc_base_flag` equal to 0 and that particular `nnpfc_id` value within the current CLVS, whichever is earlier.

`nnpfc_mode_idc`, when equal to 0, indicates that the neural network information is contained in the NNPf SEI message, and the neural network information is in the format of an ISO/IEC 15938-17 bitstream. `nnpfc_mode_idc` equal to 1 indicates that the neural network information is identified by the URI indicated by `nnpfc_uri` with the format identified by the tag URI `nnpfc_tag_uri`.

The value of `nnpfc_mode_idc` shall be in the range of 0 to 255, inclusive. Values of 2 to 255, inclusive, for `nnpfc_mode_idc` are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders conforming to this version of this document shall ignore NNPf SEI messages with `nnpfc_mode_idc` in the range of 2 to 255, inclusive.

`nnpfc_alignment_zero_bit_a` shall be equal to 0.

nnpfc_tag_uri contains a tag URI with syntax and semantics as specified in IETF RFC 4151 identifying the format and associated information about the neural network used as a base NNPF or an update relative to the base NNPF with the same **nnpfc_id** value specified by **nnpfc_uri**.

NOTE 3 **nnpfc_tag_uri** enables uniquely identifying the format of neural network data specified by **nnpfc_uri** without needing a central registration authority.

nnpfc_tag_uri equal to "tag:iso.org,2023:15938-17" indicates that the neural network data identified by **nnpfc_uri** conforms to ISO/IEC 15938-17.

nnpfc_uri contains a URI with syntax and semantics as specified in IETF Internet Standard 66 identifying the neural network used as a base NNPF or an update relative to the base NNPF with the same **nnpfc_id** value.

nnpfc_property_present_flag equal to 1 specifies that syntax elements related to the filter properties including purpose, input formatting, output formatting, and complexity are present. **nnpfc_property_present_flag** equal to 0 specifies that no syntax elements related to the filter properties are present.

When **nnpfc_base_flag** is equal to 1, **nnpfc_property_present_flag** shall be equal to 1.

When **nnpfc_property_present_flag** is equal to 0, the values of all syntax elements that may be present only when **nnpfc_property_present_flag** is equal to 1 are inferred to be equal to their corresponding syntax elements, respectively, in the NNPF SEI message that contains the base NNPF for which this SEI message provides an update.

When an NNPF SEI message **nnpfcCurr** is not the first NNPF SEI message, in decoding order, that has a particular **nnpfc_id** value within the current CLVS, is not a repetition of the first NNPF SEI message with that particular **nnpfc_id** value (in this case the value of **nnpfc_base_flag** is equal to 0), and the value of **nnpfc_property_present_flag** is equal to 1, the following constraints apply:

- The values of syntax elements following **nnpfc_property_present_flag** and preceding **nnpfc_complexity_info_present_flag**, in decoding order, in the NNPF SEI message shall be the same as the values of corresponding syntax elements in the first NNPF SEI message, in decoding order, that has that particular **nnpfc_id** value within the current CLVS.
- Either **nnpfc_complexity_info_present_flag** shall be equal to 0 or both **nnpfc_complexity_info_present_flag** shall be equal to 1 in the first NNPF SEI message, in decoding order, that has that particular **nnpfc_id** value within the current CLVS (denoted as **nnpfcBase** below) and all the following constraints apply:
- **nnpfc_parameter_type_idc** in **nnpfcCurr** shall be equal to **nnpfc_parameter_type_idc** in **nnpfcBase**.
- **nnpfc_log2_parameter_bit_length_minus3** in **nnpfcCurr**, when present, shall be less than or equal to **nnpfc_log2_parameter_bit_length_minus3** in **nnpfcBase**.
- If **nnpfc_num_parameters_idc** in **nnpfcBase** is equal to 0, **nnpfc_num_parameters_idc** in **nnpfcCurr** shall be equal to 0.
- Otherwise (**nnpfc_num_parameters_idc** in **nnpfcBase** is greater than 0), **nnpfc_num_parameters_idc** in **nnpfcCurr** shall be greater than 0 and less than or equal to **nnpfc_num_parameters_idc** in **nnpfcBase**.
- If **nnpfc_num_kmac_operations_idc** in **nnpfcBase** is equal to 0, **nnpfc_num_kmac_operations_idc** in **nnpfcCurr** shall be equal to 0.
- Otherwise (**nnpfc_num_kmac_operations_idc** in **nnpfcBase** is greater than 0), **nnpfc_num_kmac_operations_idc** in **nnpfcCurr** shall be greater than 0 and less than or equal to **nnpfc_num_kmac_operations_idc** in **nnpfcBase**.
- If **nnpfc_total_kilobyte_size** in **nnpfcBase** is equal to 0, **nnpfc_total_kilobyte_size** in **nnpfcCurr** shall be equal to 0.
- Otherwise (**nnpfc_total_kilobyte_size** in **nnpfcBase** is greater than 0), **nnpfc_total_kilobyte_size** in **nnpfcCurr** shall be greater than 0 and less than or equal to **nnpfc_total_kilobyte_size** in **nnpfcBase**.

nnpfc_num_input_pics_minus1 plus 1 specifies the number of pictures used as input for the NNPF. The value of **nnpfc_num_input_pics_minus1** shall be in the range of 0 to 63, inclusive. When **PictureRateUpsamplingFlag** is equal to 1, the value of **nnpfc_num_input_pics_minus1** shall be greater than 0.

The variable **numInputPics**, specifying the number of pictures used as input for the NNPF, is derived as follows:

$$\text{numInputPics} = \text{nnpfc_num_input_pics_minus1} + 1 \quad (76)$$

nnpfc_input_pic_filtering_flag[i] equal to 1 indicates that for the *i*-th input picture the NNPF generates a corresponding output picture. **nnpfc_input_pic_filtering_flag[i]** equal to 0 indicates that for the *i*-th input picture the NNPF does not generate a corresponding output picture. Each NNPF-generated picture is stored in the output tensor of the NNPF. When **nnpfc_num_input_pics_minus1** is equal to 0 and (**nnpfc_purpose & 0x00FF**) is not equal to 0x40, **nnpfc_input_pic_filtering_flag[0]** is inferred to be equal to 1. When **nnpfc_num_input_pics_minus1** is equal to 0 and (**nnpfc_purpose & 0x00FF**) is equal to 0x40, **nnpfc_input_pic_filtering_flag[0]** is inferred to be equal to 0. When **PictureRateUpsamplingFlag** is equal to 0, **TemporalExtrapolationFlag** is equal to 0, and **nnpfc_num_input_pics_minus1** is greater than 0, **nnpfc_input_pic_filtering_flag[i]** shall be equal to 1 for at least one value of *i* in the range of 0 to **nnpfc_num_input_pics_minus1**, inclusive.

nnpfc_absent_input_pic_zero_flag equal to 1 indicates that the NNPF expects an input picture that is not present in **CandInPicList** 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** that is not present in **CandInPicList** to be represented by the input picture **inputPicB** that is the closest to **inputPicA** in output order and is present in **CandInPicList**.

nnpfc_out_sub_c_flag specifies the values of the variables **outSubWidthC** and **outSubHeightC** when **ChromaUpsamplingFlag** is equal to 1. **nnpfc_out_sub_c_flag** equal to 1 specifies that **outSubWidthC** is equal to 1 and **outSubHeightC** is equal to 1. **nnpfc_out_sub_c_flag** equal to 0 specifies that **outSubWidthC** is equal to 2 and **outSubHeightC** is equal to 1. When **ChromaFormatIdc** is equal to 2 and **nnpfc_out_sub_c_flag** is present, the value of **nnpfc_out_sub_c_flag** shall be equal to 1.

nnpfc_out_colour_format_idc, when **ColourizationFlag** is equal to 1, specifies the colour format of the NNPF-generated pictures and consequently the values of the variables **outSubWidthC** and **outSubHeightC**. **nnpfc_out_colour_format_idc** equal to 1 specifies that the colour format of the NNPF-generated pictures is the 4:2:0 format and **outSubWidthC** and **outSubHeightC** are both equal to 2. **nnpfc_out_colour_format_idc** equal to 2 specifies that the colour format of the NNPF-generated pictures is the 4:2:2 format and **outSubWidthC** is equal to 2 and **outSubHeightC** is equal to 1. **nnpfc_out_colour_format_idc** equal to 3 specifies that the colour format of the NNPF-generated pictures is the 4:4:4 format and **outSubWidthC** and **outSubHeightC** are both equal to 1. The value of **nnpfc_out_colour_format_idc** shall not be equal to 0.

When **ChromaUpsamplingFlag** and **ColourizationFlag** are both equal to 0, **outSubWidthC** and **outSubHeightC** are inferred to be equal to **SubWidthC** and **SubHeightC**, respectively.

nnpfc_pic_width_num_minus1 plus 1 and **nnpfc_pic_width_denom_minus1** plus 1 specify the numerator and denominator, respectively, for the resampling ratio of the width of the NNPF-generated pictures relative to **CroppedWidth**. Both **nnpfc_pic_width_num_minus1** and **nnpfc_pic_width_denom_minus1** shall be in the range of 0 to 65 535, inclusive.

The value of (**nnpfc_pic_width_num_minus1** + 1) ÷ (**nnpfc_pic_width_denom_minus1** + 1) shall be in the range of 1 ÷ 16 to 16, inclusive. When **nnpfc_pic_width_num_minus1** and **nnpfc_pic_width_denom_minus1** are not present, the values of **nnpfc_pic_width_num_minus1** and **nnpfc_pic_width_denom_minus1** are both inferred to be equal to 0.

The variables **nnpfcOutputPicWidth** and **nnpfcOutputPicHeight** respectively represent the width and height of the luma sample arrays of the NNPF-generated pictures when **SpatialExtrapolationFlag** is equal to 0.

The variables **nnpfcOutputPicWidth1** and **nnpfcOutputPicHeight1** respectively represent the width and height of the luma sample arrays of the NNPF-generated pictures when **SpatialExtrapolationFlag** is equal to 1.

The variable `nnpfcOutputPicWidth` is derived as follows:

$$\text{nnpfcOutputPicWidth} = \text{Ceil}(\text{CroppedWidth} * (\text{nnpfc_pic_width_num_minus1} + 1) \div (\text{nnpfc_pic_width_denom_minus1} + 1)) \quad (77)$$

When `SpatialExtrapolationFlag` is equal to 1, `nnpfcOutputPicWidth1` is derived as follows:

$$\text{nnpfcOutputPicWidth1} = \text{nnpfcOutputPicWidth} + \text{outSubWidthC} * (\text{nnpfc_spatial_extrapolation_left_offset} + \text{nnpfc_spatial_extrapolation_right_offset}) \quad (77)$$

It is a requirement of bitstream conformance that when `SpatialExtrapolationFlag` is equal to 0, `nnpfcOutputPicWidth` shall be greater than 0 and `nnpfcOutputPicWidth % outSubWidthC` shall be equal to 0.

It is a requirement of bitstream conformance that when `SpatialExtrapolationFlag` is equal to 1, `nnpfcOutputPicWidth1` shall be greater than 0 and `nnpfcOutputPicWidth1 % outSubWidthC` shall be equal to 0.

`nnpfc_pic_height_num_minus1` plus 1 and `nnpfc_pic_height_denom_minus1` plus 1 specify the numerator and denominator, respectively, for the resampling ratio of the height of the NNPF-generated pictures relative to `CroppedHeight`. Both `nnpfc_pic_height_num_minus1` and `nnpfc_pic_height_denom_minus1` shall be in the range of 0 to 65 535, inclusive.

The value of $(\text{nnpfc_pic_height_num_minus1} + 1) \div (\text{nnpfc_pic_height_denom_minus1} + 1)$ shall be in the range of $1 \div 16$ to 16, inclusive. When `nnpfc_pic_height_num_minus1` and `nnpfc_pic_height_denom_minus1` are not present, the values of `nnpfc_pic_height_num_minus1` and `nnpfc_pic_height_denom_minus1` are both inferred to be equal to 0.

The variable `nnpfcOutputPicHeight` is derived as follows:

$$\text{nnpfcOutputPicHeight} = \text{Ceil}(\text{CroppedHeight} * (\text{nnpfc_pic_height_num_minus1} + 1) \div (\text{nnpfc_pic_height_denom_minus1} + 1)) \quad (78)$$

When `SpatialExtrapolationFlag` is equal to 1, `nnpfcOutputPicHeight1` is derived as follows:

$$\text{nnpfcOutputPicHeight1} = \text{nnpfcOutputPicHeight} + \text{outSubHeightC} * (\text{nnpfc_spatial_extrapolation_top_offset} + \text{nnpfc_spatial_extrapolation_bottom_offset}) \quad (78)$$

It is a requirement of bitstream conformance that when `SpatialExtrapolationFlag` is equal to 0, `nnpfcOutputPicHeight` shall be greater than 0 and `nnpfcOutputPicHeight % outSubHeightC` shall be equal to 0.

It is a requirement of bitstream conformance that when `SpatialExtrapolationFlag` is equal to 1, `nnpfcOutputPicHeight1` shall be greater than 0 and `nnpfcOutputPicHeight1 % outSubHeightC` shall be equal to 0.

When `ResolutionResamplingFlag` is equal to 1, at least one the following conditions shall be true:

- The value of `nnpfcOutputPicWidth` is not equal to `CroppedWidth`.
- The value of `nnpfcOutputPicHeight` is not equal to `CroppedHeight`.
- `SpatialExtrapolationFlag` is equal to 1.

`nnpfc_interpolated_pics[i]` specifies the number of interpolated pictures generated by the NNPF between the *i*-th and the (*i* + 1)-th input picture for the NNPF. The value of `nnpfc_interpolated_pics[i]` shall be in the range of 0 to 63, inclusive. When the `nnpfc_interpolated_pics[i]` syntax elements are present, the value of `nnpfc_interpolated_pics[i]` shall be greater than 0 for at least one value of *i* in the range of 0 to `nnpfc_num_input_pics_minus1` – 1, inclusive.

NOTE 4 When `PictureRateUpsamplingFlag` is equal to 1 for an NNPF and the NNPF SEI message that activated this NNPF has `nnpfa_persistence_flag` equal to 1, only for a single value of *i* in the range of 0 to `numInputPics` – 1, inclusive, the value of `nnpfc_interpolated_pics[i]` is greater than 0.

nnpfc_extrapolated_pics_minus1 plus 1 specifies the number of extrapolated pictures generated by the NNPF subsequent to all input pictures for the NNPF in output order. The value of **nnpfc_extrapolated_pics_minus1** shall be in the range of 0 to 62, inclusive.

The variables **NumInpPicsInOutputTensor**, specifying the number of pictures that have a corresponding input picture and are present in the output tensor of the NNPF, **InpIdx[idx]**, specifying the input picture index, to the list of input pictures in reverse output order, of the **idx**-th picture that is present in the output tensor of the NNPF and has a corresponding input picture, and **numPicsInOutputTensor**, specifying the total number of pictures present in the output tensor of the NNPF, are derived as follows:

```

for( i = 0, numPicsInOutputTensor = 0; i < numInputPics; i++ )
    if( nnpfc_input_pic_filtering_flag[ i ] ) {
        InpIdx[ numPicsInOutputTensor ] = i
        numPicsInOutputTensor++
    }
    }
    NumInpPicsInOutputTensor = numPicsInOutputTensor

    if( PictureRateUpsamplingFlag )
        for( i = 0; i <= numInputPics - 2; i++ )
            numPicsInOutputTensor += nnpfc_interpolated_pics[ i ]

    if( TemporalExtrapolationFlag )
        numPicsInOutputTensor += nnpfc_extrapolated_pics + 1

```

(79)

nnpfc_spatial_extrapolation_left_offset, **nnpfc_spatial_extrapolation_right_offset**, **nnpfc_spatial_extrapolation_top_offset**, and **nnpfc_spatial_extrapolation_bottom_offset** specify the spatial extrapolation area. When **nnpfc_spatial_extrapolation_left_offset**, **nnpfc_spatial_extrapolation_right_offset**, **nnpfc_spatial_extrapolation_top_offset**, and **nnpfc_spatial_extrapolation_bottom_offset** are each greater than or equal to 0, the luma samples with horizontal picture coordinates from **outSubWidthC * nnpfc_spatial_extrapolation_left_offset** to **nnpfcOutputPicWidth1 - (outSubWidthC * nnpfc_spatial_extrapolation_right_offset)** and vertical picture coordinates from **outSubHeightC * nnpfc_spatial_extrapolation_top_offset** to **nnpfcOutputPicHeight1 - (outSubHeightC * nnpfc_spatial_extrapolation_bottom_offset)** correspond to the spatial area of the input picture. The value of **nnpfc_spatial_extrapolation_left_offset**, **nnpfc_spatial_extrapolation_right_offset**, **nnpfc_spatial_extrapolation_top_offset** and **nnpfc_spatial_extrapolation_bottom_offset** shall be in the range of -65 536 to 65 536, inclusive. At least one of **nnpfc_spatial_extrapolation_left_offset**, **nnpfc_spatial_extrapolation_right_offset**, **nnpfc_spatial_extrapolation_top_offset** and **nnpfc_spatial_extrapolation_bottom_offset** shall be greater than 0.

nnpfc_component_last_flag equal to 1 indicates that the last dimension in the input tensor **inputTensor** to the NNPF and the output tensor **outputTensor** of the NNPF is used for a current channel. **nnpfc_component_last_flag** equal to 0 indicates that the third dimension in the input tensor **inputTensor** to the NNPF and the output tensor **outputTensor** of the NNPF is used for a current channel.

NOTE 5 The first dimension in the input tensor and in the output tensor is used for the batch index, which is a common practice in some neural network frameworks. While the pseudocode in the semantics of this SEI message use the batch size corresponding to the batch index equal to 0, it is up to the post-processing implementation to determine the batch size used as the input to the neural network inference process.

NOTE 6 For example, when **nnpfc_inp_order_idc** is equal to 3 and **nnpfc_auxiliary_inp_idc** is equal to 1, there are 7 channels in the input tensor, including four luma matrices, two chroma matrices, and one auxiliary input matrix. In this case, the process **DeriveInputTensors()** would derive each of these 7 channels of the input tensor one by one, and when a particular channel of these channels is processed, that channel is referred to as the current channel during the process.

nnpfc_inp_format_idc indicates the method of converting a sample value of the input picture to an input value to the NNPF. The value of **nnpfc_inp_format_idc** shall be in the range of 0 to 255, inclusive. Values of **nnpfc_inp_format_idc** in the range of 2 to 255, inclusive, are reserved for future specification by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders conforming to this version of this document shall ignore NNPF SEI messages with **nnpfc_inp_format_idc** in the range of 2 to 255, inclusive.

When **nnpfc_inp_format_idc** is equal to 0, the input values to the NNPF are real numbers and the functions **InpY()** and **InpC()** are specified as follows:

$$\text{InpY}(x) = x \div ((1 \ll \text{BitDepth}_Y) - 1) \quad (80)$$

$$\text{InpC}(x) = x \div ((1 \ll \text{BitDepth}_C) - 1) \quad (81)$$

When **nnpfc_inp_format_idc** is equal to 1, the input values to the NNPF are unsigned integer numbers and the functions **InpY()** and **InpC()** are specified as follows:

shiftY = **BitDepth_Y** – **inpTensorBitDepth_Y**

if(**inpTensorBitDepth_Y** >= **BitDepth_Y**)

$$\text{InpY}(x) = x \ll (\text{inpTensorBitDepth}_Y - \text{BitDepth}_Y) \quad (82)$$

else

$$\text{InpY}(x) = \text{Clip3}(0, (1 \ll \text{inpTensorBitDepth}_Y) - 1, (x + (1 \ll (\text{shiftY} - 1))) \gg \text{shiftY})$$

shiftC = **BitDepth_C** – **inpTensorBitDepth_C**

$$\text{InpC}(x) = x \ll (\text{inpTensorBitDepth}_C - \text{BitDepth}_C) \quad (83)$$

else

$$\text{InpC}(x) = \text{Clip3}(0, (1 \ll \text{inpTensorBitDepth}_C) - 1, (x + (1 \ll (\text{shiftC} - 1))) \gg \text{shiftC})$$

The variable **inpTensorBitDepth_Y** is derived from the syntax element **nnpfc_inp_tensor_luma_bitdepth_minus8** as specified below. The variable **inpTensorBitDepth_C** is derived from the syntax element **nnpfc_inp_tensor_chroma_bitdepth_minus8** as specified below.

nnpfc_auxiliary_inp_idc greater than 0 indicates that auxiliary input data is present in the input tensor of the NNPF. **nnpfc_auxiliary_inp_idc** equal to 0 indicates that auxiliary input data is not present in the input tensor. **nnpfc_auxiliary_inp_idc** in the range of 1 to 7, inclusive, specifies that auxiliary input data is derived as specified in pseudocode statement 83.

The value of **nnpfc_auxiliary_inp_idc** shall be in the range of 0 to 255, inclusive. Values of 8 to 255, inclusive, for **nnpfc_auxiliary_inp_idc** are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders conforming to this version of this document shall ignore NNPF SEI messages with **nnpfc_auxiliary_inp_idc** in the range of 8 to 255, inclusive.

When (**nnpfc_auxiliary_inp_idc** & 1) is equal to 1 the auxiliary input data includes **strengthControlScaledVal[i]**.

When (**nnpfc_auxiliary_inp_idc** & 2) is equal to 2 the auxiliary input data includes text prompt character values.

When (**nnpfc_auxiliary_inp_idc** & 4) is equal to 4, the auxiliary input data includes a seed value.

The variable **nnpfcNumAuxMatrices**, specifying the number of auxiliary input matrices in the input tensor, is set equal to (**nnpfc_auxiliary_inp_idc** & 1) + ((**nnpfc_auxiliary_inp_idc** & 2) >> 1) + ((**nnpfc_auxiliary_inp_idc** & 4) >> 2).

nnpfc_inband_prompt_flag equal to 1 specifies that the text prompt string to be included in the input tensor is included in this NNPF SEI message or in an NNPF SEI message activating the NNPF defined by

this NNPFC SEI message. `nnpfc_inband_prompt_flag` equal to 0 specifies that the text prompt string to be included in the input tensor is provided to the decoding system by external means.

`nnpfc_alignment_zero_bit_c` shall be equal to 0.

`nnpfc_prompt` specifies the text string prompt used as input for an NNPF, for example for generating the contents of the spatial extrapolation image area.

When `nnpfc_inband_prompt_flag` is equal to 1, it is a requirement of bitstream conformance that `nnpfc_prompt` shall not be a null string or `nnpfa_prompt_update_flag` in all NNPF SEI messages that activate the NNPF defined by this NNPFC SEI message shall be equal to 1.

The variable `nnpfcPrompt`, specifying the text prompt string to be provided in the input tensor for a particular picture `picA` for which the NNPF is activated, is derived as follows:

- If `nnpfc_inband_prompt_flag` is equal to 1 and `nnpfa_prompt_update_flag` is equal to 1, `nnpfcPrompt` is set equal to `nnpfa_prompt`.
- Otherwise, if `nnpfc_inband_prompt_flag` is equal to 1 and `nnpfa_prompt_update_flag` is equal to 0, `nnpfcPrompt` is set equal to `nnpfc_prompt`.
- Otherwise, if `nnpfc_inband_prompt_flag` is equal to 0 and a text prompt string is provided by external means, `nnpfcPrompt` is set equal to that text prompt string.
- Otherwise (`nnpfc_inband_prompt_flag` is equal to 0 and no text prompt string is provided by external means), `nnpfcPrompt` is set equal to a null string.

`nnpfc_inband_seed_flag` equal to 1 specifies that the seed value to be included in the input tensor is included in this NNPFC SEI message or in an NNPF SEI message activating the NNPF defined by this NNPFC SEI message. `nnpfc_inband_seed_flag` equal to 0 specifies that the seed value to be included in the input tensor is provided to the decoding system by external means.

`nnpfc_seed` specifies the seed value used as input for an NNPF.

NOTE 7 An NNPF can use the `nnpfcSeedVal` to initialize a pseudorandom number generator. When `nnpfc_inband_seed_flag` is equal to 1, an encoder can control that all inference engines for the NNPF use the same seed value and therefore reach the same NNPF-generated pictures.

The variable `nnpfcSeedVal`, specifying the seed value to be provided in the input tensor for a particular picture `picA` for which the NNPF is activated, is derived as follows:

- If `nnpfc_inband_seed_flag` is equal to 1 and `nnpfa_seed_update_flag` is equal to 1, `nnpfcSeedVal` is set equal to `nnpfa_seed`.
- Otherwise, if `nnpfc_inband_seed_flag` is equal to 1 and `nnpfa_seed_update_flag` is equal to 0, `nnpfcSeedVal` is set equal to `nnpfc_seed`.
- Otherwise, if `nnpfc_inband_seed_flag` is equal to 0 and a seed value is provided by external means, `nnpfcSeedVal` is set equal to that seed value.
- Otherwise (`nnpfc_inband_seed_flag` is equal to 0 and no seed value is provided by external means), `nnpfcSeedVal` is set equal to 0.

`nnpfc_inp_order_idc` indicates the method of ordering the sample arrays of an input picture to form an input tensor to the NNPF.

The value of `nnpfc_inp_order_idc` shall be in the range of 0 to 255, inclusive. Values of 4 to 255, inclusive, for `nnpfc_inp_order_idc` are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders conforming to this version of this document shall ignore NNPFC SEI messages with `nnpfc_inp_order_idc` in the range of 4 to 255, inclusive.

When `ChromaFormatIdc` is not equal to 1, `nnpfc_inp_order_idc` shall not be equal to 3.

When ChromaFormatIdc is equal to 0, nnpfc_inp_order_idc shall be equal to 0.

When ChromaUpsamplingFlag is equal to 1, nnpfc_inp_order_idc shall not be equal to 0.

Table 21 contains an informative description of nnpfc_inp_order_idc values.

Table 21 — Description of nnpfc_inp_order_idc values

nnpfc_inp_order_idc	Description
0	One luma matrix is present in the input tensor, and the number of channels is 1 + nnpfcNumAuxMatrices.
1	Two chroma matrices are present in the input tensor, and the number of channels is 2 + nnpfcNumAuxMatrices.
2	One luma and two chroma matrices are present in the input tensor, and the number of channels is 3 + nnpfcNumAuxMatrices.
3	Four luma matrices and two chroma matrices are present in the input tensor, and the number of channels is 6 + nnpfcNumAuxMatrices. The luma channels are derived in an interleaved manner as illustrated in Figure 12. This nnpfc_inp_order_idc can only be used when the input chroma format is 4:2:0.
4..255	Reserved

nnpfc_inp_tensor_luma_bitdepth_minus8 plus 8 specifies the bit depth of luma sample values in the input integer tensor. The value of $\text{inpTensorBitDepth}_Y$ is derived as follows:

$$\text{inpTensorBitDepth}_Y = \text{nnpfc_inp_tensor_luma_bitdepth_minus8} + 8 \quad (84)$$

It is a requirement of bitstream conformance that the value of **nnpfc_inp_tensor_luma_bitdepth_minus8** shall be in the range of 0 to 24, inclusive.

nnpfc_inp_tensor_chroma_bitdepth_minus8 plus 8 specifies the bit depth of chroma sample values in the input integer tensor. The value of $\text{inpTensorBitDepth}_C$ is derived as follows:

$$\text{inpTensorBitDepth}_C = \text{nnpfc_inp_tensor_chroma_bitdepth_minus8} + 8 \quad (85)$$

It is a requirement of bitstream conformance that the value of **nnpfc_inp_tensor_chroma_bitdepth_minus8** shall be in the range of 0 to 24, inclusive.

nnpfc_out_format_idc equal to 0 indicates that the sample values output by the NNPF are real numbers where the value range of 0 to 1, inclusive, maps linearly to the unsigned integer value range of 0 to $(1 \ll \text{bitDepth}) - 1$, inclusive, for any desired bit depth bitDepth for subsequent post-processing or displaying.

nnpfc_out_format_idc equal to 1 indicates that the luma sample values output by the NNPF are unsigned integer numbers in the range of 0 to $(1 \ll \text{outTensorBitDepth}_Y) - 1$, inclusive, and the chroma sample values output by the NNPF are unsigned integer numbers in the range of 0 to $(1 \ll \text{outTensorBitDepth}_C) - 1$, inclusive.

The value of **nnpfc_out_format_idc** shall be in the range of 0 to 255, inclusive. Values of 2 to 255, inclusive, for **nnpfc_out_format_idc** are reserved for future specification by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders conforming to this version of this document shall ignore NNPF SEI messages with **nnpfc_out_format_idc** in the range of 2 to 255, inclusive.

nnpfc_out_order_idc indicates the output order of samples resulting from the NNPF.

The value of **nnpfc_out_order_idc** shall be in the range of 0 to 255, inclusive. Values of 4 to 255, inclusive, for **nnpfc_out_order_idc** are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders conforming to this version of this document shall ignore NNPF SEI messages with **nnpfc_out_order_idc** in the range of 4 to 255, inclusive.

When ChromaUpsamplingFlag is equal to 1, **nnpfc_out_order_idc** shall not be equal to 0 or 3.

When ColourizationFlag is equal to 1, nnpfc_out_order_idc shall not be equal to 0.

Table 22 contains an informative description of nnpfc_out_order_idc values.

Table 22 — Description of nnpfc_out_order_idc values

nnpfc_out_order_idc	Description
0	Only the luma matrix is present in the output tensor, thus the number of channels is 1.
1	Only the chroma matrices are present in the output tensor, thus the number of channels is 2.
2	The luma and chroma matrices are present in the output tensor, thus the number of channels is 3.
3	Four luma matrices and two chroma matrices are present in the output tensor, thus the number of channels is 6. This nnpfc_out_order_idc can only be used when the output chroma format is 4:2:0.
4..255	Reserved

nnpfc_out_tensor_luma_bitdepth_minus8 plus 8 specifies the bit depth of luma sample values in the output integer tensor. The value of nnpfc_out_tensor_luma_bitdepth_minus8 shall be in the range of 0 to 24, inclusive. The value of outTensorBitDepth_Y is derived as follows:

$$\text{outTensorBitDepth}_Y = \text{nnpfc_out_tensor_luma_bitdepth_minus8} + 8 \quad (86)$$

nnpfc_out_tensor_chroma_bitdepth_minus8 plus 8 specifies the bit depth of chroma sample values in the output integer tensor. The value of nnpfc_out_tensor_chroma_bitdepth_minus8 shall be in the range of 0 to 24, inclusive. The value of outTensorBitDepth_C is derived as follows:

$$\text{outTensorBitDepth}_C = \text{nnpfc_out_tensor_chroma_bitdepth_minus8} + 8 \quad (87)$$

When BitDepthUpsamplingFlag is equal to 1, the value of nnpfc_out_format_idc shall be equal to 1 and at least one of the following conditions shall be true:

- nnpfc_out_tensor_luma_bitdepth_minus8 is present and outTensorBitDepth_Y is greater than BitDepth_Y.
- nnpfc_out_tensor_chroma_bitdepth_minus8 is present and outTensorBitDepth_C is greater than BitDepth_C.

When nnpfc_inp_tensor_luma_bitdepth_minus8, nnpfc_inp_tensor_chroma_bitdepth_minus8, nnpfc_out_tensor_luma_bitdepth_minus8, and nnpfc_out_tensor_chroma_bitdepth_minus8 are present and outTensorBitDepth_Y is greater than inpTensorBitDepth_Y, outTensorBitDepth_C shall not be less than inpTensorBitDepth_C. When nnpfc_inp_tensor_luma_bitdepth_minus8, nnpfc_inp_tensor_chroma_bitdepth_minus8, nnpfc_out_tensor_luma_bitdepth_minus8, and nnpfc_out_tensor_chroma_bitdepth_minus8 are present and outTensorBitDepth_C is greater than inpTensorBitDepth_C, outTensorBitDepth_Y shall not be less than inpTensorBitDepth_Y.

nnpfc_separate_colour_description_present_flag equal to 1 indicates that a distinct combination of colour primaries, transfer characteristics, matrix coefficients, and scaling and offset values applied in association with the matrix coefficients for the picture resulting from the NNPF is specified in the SEI message syntax structure. nnpfc_separate_colour_description_present_flag equal to 0 indicates that the combination of colour primaries, transfer characteristics, matrix coefficients, and scaling and offset values applied in association with the matrix coefficients for the picture resulting from the NNPF is the same as implied by the VUI parameters vui_colour_primaries, vui_transfer_characteristics, vui_matrix_coeffs, and vui_full_range_flag that are indicated or inferred for the CLVS.

nnpfc_colour_primaries has the same semantics as specified in clause 7.3 for the vui_colour_primaries syntax element, except as follows:

- nnpfc_colour_primaries specifies the colour primaries of the picture resulting from applying the NNPF specified in the SEI message, rather than the colour primaries used for the CLVS.
- When nnpfc_colour_primaries is not present in the NNPF SEI message, the value of nnpfc_colour_primaries is inferred to be equal to vui_colour_primaries.

nnpfc_transfer_characteristics has the same semantics as specified in clause 7.3 for the `vui_transfer_characteristics` syntax element, except as follows:

- `nnpfc_transfer_characteristics` specifies the transfer characteristics of the picture resulting from applying the NNPF specified in the SEI message, rather than the transfer characteristics used for the CLVS.
- When `nnpfc_transfer_characteristics` is not present in the NNPF SEI message, the value of `nnpfc_transfer_characteristics` is inferred to be equal to `vui_transfer_characteristics`.

nnpfc_matrix_coeffs describes the pseudocode steps used in deriving luma and chroma signals from the green, blue, and red, or Y, Z, and X primaries. Its semantics apply to the pictures resulting from applying the NNPF specified in this SEI message and are as specified for `MatrixCoefficients` in Rec. ITU-T H.273 | ISO/IEC 23091-2 with `BitDepthY` and `BitDepthC` being equal to `outTensorBitDepthY` and `outTensorBitDepthC`, respectively.

When `nnpfc_matrix_coeffs` is not present in the NNPF SEI message, the value of `nnpfc_matrix_coeffs` is inferred to be equal to `vui_matrix_coeffs`.

`nnpfc_matrix_coeffs` shall not be equal to 0, 16, or 17 unless both of the following conditions are true:

- Both `nnpfc_out_tensor_chroma_bitdepth_minus8` and `nnpfc_out_tensor_luma_bitdepth_minus8` are present, and `nnpfc_out_tensor_chroma_bitdepth_minus8` is equal to `nnpfc_out_tensor_luma_bitdepth_minus8`.
- Both `nnpfc_out_tensor_chroma_bitdepth_minus8` and `nnpfc_out_tensor_luma_bitdepth_minus8` are present, and `nnpfc_out_order_idc` is equal to 2, `outSubHeightC` is equal to 1, and `outSubWidthC` is equal to 1.

NOTE 8 When both of the above two conditions are true and `vui_matrix_coeffs` is equal to 0, `nnpfc_matrix_coeffs` is expected to be present.

`nnpfc_matrix_coeffs` shall not be equal to 8 unless one of the following conditions is true:

- Both `nnpfc_out_tensor_chroma_bitdepth_minus8` and `nnpfc_out_tensor_luma_bitdepth_minus8` are present; and `nnpfc_out_tensor_chroma_bitdepth_minus8` is equal to `nnpfc_out_tensor_luma_bitdepth_minus8`.
- Both `nnpfc_out_tensor_chroma_bitdepth_minus8` and `nnpfc_out_tensor_luma_bitdepth_minus8` are present; and `nnpfc_out_tensor_chroma_bitdepth_minus8` is equal to `nnpfc_out_tensor_luma_bitdepth_minus8` + 1, `nnpfc_out_order_idc` is equal to 2, `outSubHeightC` is equal to 1, and `outSubWidthC` is equal to 1.

NOTE 9 When one of the above two conditions is true and `vui_matrix_coeffs` is equal to 8, `nnpfc_matrix_coeffs` is expected to be present.

nnpfc_full_range_flag indicates the scaling and offset values applied in association with the matrix coefficients as specified by `nnpfc_matrix_coeffs`. Its semantics are as specified for the `VideoFullRangeFlag` parameter in Rec. ITU-T H.273 | ISO/IEC 23091-2. When `nnpfc_full_range_flag` is not present and `nnpfc_out_format_idc` is equal to 1, the value of `nnpfc_full_range_flag` is inferred to be equal to `vui_full_range_flag`.

nnpfc_chroma_loc_info_present_flag equal to 1 indicates the presence of the `nnpfc_chroma_sample_loc_type_frame` syntax element in the NNPF SEI message. `nnpfc_chroma_loc_info_present_flag` equal to 0 indicates the absence of the `nnpfc_chroma_sample_loc_type_frame` syntax element in the NNPF SEI message. When `nnpfc_chroma_loc_info_present_flag` is not present, its value is inferred to be equal to 0. When `ColourizationFlag` is equal to 0 or `nnpfc_out_colour_format_idc` is not equal to 1, the value of `nnpfc_chroma_loc_info_present_flag` shall be equal to 0.

nnpfc_chroma_sample_loc_type_frame, when not equal to 6 and `nnpfc_out_colour_format_idc` is equal to 1, specifies the location of chroma samples of the output pictures, as shown in Figure 1. `nnpfc_chroma_sample_loc_type_frame` equal to 6 and `nnpfc_out_colour_format_idc` equal to 1 indicates that the location of the chroma samples is unknown or unspecified or specified by other means not specified in this document. The value of `nnpfc_chroma_sample_loc_type_frame` shall be in the range of 0 to 6, inclusive.

nnpfc_overlap indicates the overlapping horizontal and vertical sample counts of adjacent input tensors of the NNPF. The value of `nnpfc_overlap` shall be in the range of 0 to 16 383, inclusive. When `SpatialExtrapolationFlag` is equal to 1, `nnpfc_overlap` is inferred to be equal to 0.

nnpfc_constant_patch_size_flag equal to 1 indicates that the NNPF accepts exactly the patch size indicated by **nnpfc_patch_width_minus1** and **nnpfc_patch_height_minus1** as input. **nnpfc_constant_patch_size_flag** equal to 0 indicates that the NNPF accepts as input any patch size with width **inpPatchWidth** and height **inpPatchHeight** such that the width of an extended patch (i.e., a patch plus the overlapping area), which is equal to $\text{inpPatchWidth} + 2 * \text{nnpfc_overlap}$, is a positive integer multiple of **nnpfc_extended_patch_width_cd_delta_minus1 + 1 + 2 * nnpfc_overlap, and the height of the extended patch, which is equal to $\text{inpPatchHeight} + 2 * \text{nnpfc_overlap}$, is a positive integer multiple of **nnpfc_extended_patch_height_cd_delta_minus1 + 1 + 2 * nnpfc_overlap**. When **SpatialExtrapolationFlag** is equal to 1, **nnpfc_constant_patch_size_flag** is inferred to be equal to 1.**

nnpfc_patch_width_minus1 plus 1, when **nnpfc_constant_patch_size_flag** equal to 1, indicates the horizontal sample counts of the patch size required for the input to the NNPF. The value of **nnpfc_patch_width_minus1** shall be in the range of 0 to $\text{Min}(32\,766, \text{CroppedWidth} - 1)$, inclusive.

nnpfc_patch_height_minus1 plus 1, when **nnpfc_constant_patch_size_flag** equal to 1, indicates the vertical sample counts of the patch size required for the input to the NNPF. The value of **nnpfc_patch_height_minus1** shall be in the range of 0 to $\text{Min}(32\,766, \text{CroppedHeight} - 1)$, inclusive.

nnpfc_extended_patch_width_cd_delta_minus1 plus 1 plus $2 * \text{nnpfc_overlap}$, when **nnpfc_constant_patch_size_flag** equal to 0, indicates a common divisor of all allowed values of the width of an extended patch required for the input to the NNPF. The value of **nnpfc_extended_patch_width_cd_delta_minus1** shall be in the range of 0 to $\text{Min}(32\,766, \text{CroppedWidth} - 1)$, inclusive.

nnpfc_extended_patch_height_cd_delta_minus1 plus 1 plus $2 * \text{nnpfc_overlap}$, when **nnpfc_constant_patch_size_flag** equal to 0, indicates a common divisor of all allowed values of the height of an extended patch required for the input to the NNPF. The value of **nnpfc_extended_patch_height_cd_delta_minus1** shall be in the range of 0 to $\text{Min}(32\,766, \text{CroppedHeight} - 1)$, inclusive.

Let the variables **inpPatchWidth** and **inpPatchHeight** be the patch size width and the patch size height, respectively.

If **nnpfc_constant_patch_size_flag** is equal to 0, the following applies:

- The values of **inpPatchWidth** and **inpPatchHeight** are either provided by external means not specified in this document or set by the post-processor itself.
- The value of $\text{inpPatchWidth} + 2 * \text{nnpfc_overlap}$ shall be a positive integer multiple of **nnpfc_extended_patch_width_cd_delta_minus1 + 1 + 2 * nnpfc_overlap** and **inpPatchWidth** shall be less than or equal to **CroppedWidth**. The value of $\text{inpPatchHeight} + 2 * \text{nnpfc_overlap}$ shall be a positive integer multiple of **nnpfc_extended_patch_height_cd_delta_minus1 + 1 + 2 * nnpfc_overlap** and **inpPatchHeight** shall be less than or equal to **CroppedHeight**.

Otherwise (**nnpfc_constant_patch_size_flag** is equal to 1), the value of **inpPatchWidth** is set equal to **nnpfc_patch_width_minus1 + 1** and the value of **inpPatchHeight** is set equal to **nnpfc_patch_height_minus1 + 1**.

The variables **outPatchWidth**, **outPatchHeight**, **horCScaling**, **verCScaling**, **outPatchCWidth**, and **outPatchCHeight** are derived as follows:

$$\text{outPatchWidth} = (\text{nnpfcOutputPicWidth} * \text{inpPatchWidth}) / \text{CroppedWidth} \quad (88)$$

$$\text{outPatchHeight} = (\text{nnpfcOutputPicHeight} * \text{inpPatchHeight}) / \text{CroppedHeight} \quad (89)$$

$$\text{horCScaling} = \text{SubWidthC} / \text{outSubWidthC} \quad (90)$$

$$\text{verCScaling} = \text{SubHeightC} / \text{outSubHeightC} \quad (91)$$

$$\text{outPatchCWidth} = \text{outPatchWidth} * \text{horCScaling} \quad (92)$$

$$\text{outPatchCHeight} = \text{outPatchHeight} * \text{verCScaling} \quad (93)$$

It is a requirement of bitstream conformance that when `SpatialExtrapolationFlag` is equal to 0, `outPatchWidth * CroppedWidth` shall be equal to `nnpfcOutputPicWidth * inpPatchWidth` and `outPatchHeight * CroppedHeight` shall be equal to `nnpfcOutputPicHeight * inpPatchHeight`.

It is a requirement of bitstream conformance that when `SpatialExtrapolationFlag` is equal to 1, `outPatchWidth * CroppedWidth` shall be equal to `nnpfcOutputPicWidth1 * inpPatchWidth` and `outPatchHeight * CroppedHeight` shall be equal to `nnpfcOutputPicHeight1 * inpPatchHeight`.

nnpfc_padding_type indicates the process of padding when referencing sample locations outside the boundaries of the input picture as described in Table 23. The value of `nnpfc_padding_type` shall be in the range of 0 to 15, inclusive. Values of 5 to 15, inclusive, for `nnpfc_padding_type` are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this version of this document. Decoders conforming to this version of this document shall ignore NNPFC SEI messages with `nnpfc_padding_type` in the range of 5 to 15, inclusive.

Table 23 — Informative description of `nnpfc_padding_type` values

<code>nnpfc_padding_type</code>	Description
0	Zero padding
1	Replication padding
2	Reflection padding
3	Wrap-around padding
4	Fixed padding
5..15	reserved

nnpfc_luma_padding_val indicates the luma value to be used for padding when `nnpfc_padding_type` is equal to 4. The value of `nnpfc_luma_padding_val` shall be in the range of 0 to $(1 \ll \text{BitDepth}_Y) - 1$, inclusive.

nnpfc_cb_padding_val indicates the Cb value to be used for padding when `nnpfc_padding_type` is equal to 4. The value of `nnpfc_cb_padding_val` shall be in the range of 0 to $(1 \ll \text{BitDepth}_C) - 1$, inclusive.

nnpfc_cr_padding_val indicates the Cr value to be used for padding when `nnpfc_padding_type` is equal to 4. The value of `nnpfc_cr_padding_val` shall be in the range of 0 to $(1 \ll \text{BitDepth}_C) - 1$, inclusive.

The function `InpSampleVal( y, x, picHeight, picWidth, croppedPic, cldx )` with inputs being a vertical sample location `y`, a horizontal sample location `x`, a picture height `picHeight`, a picture width `picWidth`, sample array `croppedPic`, and component index `cldx` (equal to 0 for luma, 1 for Cb, and 2 for Cr) returns the value of `sampleVal` derived as follows:

NOTE 10 For the inputs to the function `InpSampleVal( )`, the vertical location is listed before the horizontal location for compatibility with input tensor conventions of some inference engines.

```

if( nnpfc_padding_type == 0 )
    if( y < 0 || x < 0 || y >= picHeight || x >= picWidth )
        sampleVal = 0
    else
        sampleVal = croppedPic[ x ][ y ]
else if( nnpfc_padding_type == 1 )
    sampleVal = croppedPic[ Clip3( 0, picWidth - 1, x ) ][ Clip3( 0, picHeight - 1, y ) ]

```

(94)