



International
Standard

ISO/IEC 23002-7

**Information technology — MPEG
video technologies —**

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

Technologies de l'information — Technologies vidéo MPEG —

*Partie 7: Messages d'améliorations complémentaires polyvalents
pour les flux binaires vidéo codés*

**Third edition
2024-10**

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

Full document details: ISO IEC 23002-7-2024

 **Documents**
© ISO/IEC 2024 Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

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

Contents

Page

Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	2
4 Abbreviated terms	8
5 Conventions	9
5.1 General	9
5.2 Arithmetic operators	9
5.3 Logical operators	10
5.4 Relational operators	10
5.5 Bit-wise operators	10
5.6 Assignment operators	11
5.7 Range notation	11
5.8 Mathematical functions	11
5.9 Order of operation precedence	13
5.10 Variables, syntax elements and tables	13
5.11 Text description of logical operations	14
5.12 Processes	16
6 Syntax and semantics	16
6.1 General	16
6.2 Method of specifying syntax in tabular form	17
6.3 Specification of syntax functions and descriptors	18
7 Video usability information parameters	19
7.1 General	19
7.2 VUI parameters syntax	19
7.3 VUI parameters semantics	20
8 SEI messages	27
8.1 General	27
8.2 Filler payload SEI message	28
8.2.1 Filler payload SEI message syntax	28
8.2.2 Filler payload SEI message semantics	28
8.3 User data registered by Rec. ITU-T T.35 SEI message	29
8.3.1 User data registered by Rec. ITU-T T.35 SEI message syntax	29
8.3.2 User data registered by Rec. ITU-T T.35 SEI message semantics	29
8.4 User data unregistered SEI message	29
8.4.1 User data unregistered SEI message syntax	29
8.4.2 User data unregistered SEI message semantics	29
8.5 Film grain characteristics SEI message	30
8.5.1 Film grain characteristics SEI message syntax	30
8.5.2 Film grain characteristics SEI message semantics	31
8.6 Frame packing arrangement SEI message	38
8.6.1 Frame packing arrangement SEI message syntax	38
8.6.2 Frame packing arrangement SEI message semantics	39
8.7 Parameter sets inclusion indication SEI message	46
8.7.1 Parameter sets inclusion indication SEI message syntax	46
8.7.2 Parameter sets inclusion indication SEI message semantics	46
8.8 Decoded picture hash SEI message	47
8.8.1 Decoded picture hash SEI message syntax	47
8.8.2 Decoded picture hash SEI message semantics	47
8.9 Mastering display colour volume SEI message	49
8.9.1 Mastering display colour volume SEI message syntax	49

8.9.2	Mastering display colour volume SEI message semantics	49
8.10	Content light level information SEI message	51
8.10.1	Content light level information SEI message syntax	51
8.10.2	Content light level information SEI message semantics	51
8.11	Dependent random access point indication SEI message	52
8.11.1	Dependent random access point indication SEI message syntax	52
8.11.2	Dependent random access point indication SEI message semantics	52
8.12	Alternative transfer characteristics information SEI message	53
8.12.1	Alternative transfer characteristics information SEI message syntax	53
8.12.2	Alternative transfer characteristics SEI message semantics	53
8.13	Ambient viewing environment SEI message	53
8.13.1	Ambient viewing environment SEI message syntax	53
8.13.2	Ambient viewing environment SEI message semantics	53
8.14	Content colour volume SEI message	54
8.14.1	Content colour volume SEI message syntax	54
8.14.2	Content colour volume SEI message semantics	54
8.15	Omnidirectional video specific SEI messages	57
8.15.1	Sample location remapping process	57
8.15.2	Equirectangular projection SEI message	67
8.15.3	Generalized cubemap projection SEI message	70
8.15.4	Sphere rotation SEI message	76
8.15.5	Region-wise packing SEI message	77
8.15.6	Omnidirectional viewport SEI message	84
8.16	Frame-field information SEI message	86
8.16.1	Frame-field information SEI message syntax	86
8.16.2	Frame-field information SEI message semantics	86
8.17	Sample aspect ratio information SEI message	89
8.17.1	Sample aspect ratio information SEI message syntax	89
8.17.2	Sample aspect ratio information SEI message semantics	89
8.18	Annotated regions SEI message	90
8.18.1	Annotated regions SEI message syntax	90
8.18.2	Annotated regions SEI message semantics	91
8.19	Scalability dimension information SEI message	94
8.19.1	Scalability dimension information SEI message syntax	94
8.19.2	Scalability dimension information SEI message semantics	95
8.20	Multiview acquisition information SEI message	96
8.20.1	Multiview acquisition information SEI message syntax	96
8.20.2	Multiview acquisition information SEI message semantics	97
8.21	Multiview view position SEI message	101
8.21.1	Multiview view position SEI message syntax	101
8.21.2	Multiview view position SEI message semantics	101
8.22	Depth representation information SEI message	102
8.22.1	Depth representation information SEI message syntax	102
8.22.2	Depth representation information SEI message semantics	102
8.23	Alpha channel information SEI message	106
8.23.1	Alpha channel information SEI message syntax	106
8.23.2	Alpha channel information SEI message semantics	106
8.24	Extended DRAP indication SEI message	109
8.24.1	Extended DRAP indication SEI message syntax	109
8.24.2	Extended DRAP indication SEI message semantics	109
8.25	Display orientation SEI message	110
8.25.1	Display orientation SEI message syntax	110
8.25.2	Display orientation SEI message semantics	111
8.26	Colour transform information SEI message	112
8.26.1	Colour transform information SEI message syntax	112
8.26.2	Colour transform information SEI message semantics	112
8.27	Shutter interval information SEI message	116
8.27.1	Shutter interval information SEI message syntax	116
8.27.2	Shutter interval information SEI message semantics	116

8.28	Neural-network post-filter SEI messages.....	117
8.28.1	General post-processing filtering process using NNPFs.....	117
8.28.2	Neural-network post-filter characteristics SEI message.....	118
8.28.3	Neural-network post-filter activation SEI message.....	141
8.29	Phase indication SEI message.....	143
8.29.1	Phase indication SEI message syntax.....	143
8.29.2	Phase indication SEI message semantics.....	143
8.30	Reserved SEI message.....	145
8.30.1	Reserved SEI message syntax.....	145
8.30.2	Reserved SEI message semantics.....	145
9	Parsing process for k-th order Exp-Golomb codes.....	145
9.1	General.....	145
9.2	Mapping process for signed Exp-Golomb codes.....	146
	Bibliography.....	148

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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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. In the IEC, see www.iec.ch/understanding-standards.

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 (as Rec. ITU-T H.274).

This third edition cancels and replaces the second edition (ISO/IEC 23002-7:2022), which has been technically revised.

The main changes are as follows:

- the addition of the shutter interval information SEI message,
- the addition of the neural-network post-filter characteristics SEI message,
- the addition of the neural-network post-filter activation SEI message, and
- the addition of the phase indication SEI message.

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

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 and www.iec.ch/national-committees.

Introduction

Versions of this document

Rec. ITU-T H.274 | ISO/IEC 23002-7 version 1 refers to the first approved version of this document. The first edition published by ITU-T as Rec. ITU-T H.274 (08/2020) and by ISO/IEC as ISO/IEC 23002-7:2021 corresponded to the first version.

Rec. ITU-T H.274 | ISO/IEC 23002-7 version 2 refers to the integrated text containing nine additional SEI messages, namely the annotated regions SEI message, the alpha channel information SEI message, the depth representation information SEI message, the multiview acquisition information SEI message, the multiview view position SEI message, the scalability dimension information SEI message, the extended dependent random access point indication SEI message, the display orientation SEI message, and the colour transform information SEI message. Besides these additional SEI messages, this version also contains corrections to various minor defects in the prior content of the specification. The second edition published by ITU-T as Rec. ITU-T H.274 (05/2022) and by ISO/IEC as ISO/IEC 23002-7:2022 corresponds to the second version.

Rec. ITU-T H.274 | ISO/IEC 23002-7 version 3 (the current version) refers to the integrated text containing four additional SEI messages, namely the shutter interval information SEI message, the neural-network post-filter characteristics SEI message, the neural-network post-filter activation SEI message, and the phase indication SEI message. Besides these additional SEI messages, this version also contains corrections to various minor defects in the prior content of the specification. The third edition published by ISO/IEC as ISO/IEC 23002-7:2024 corresponds to the third version. This third edition of ISO/IEC 23002-7 corresponds to the third edition published by ITU-T as Rec. ITU-T H.274 (09/2023).

Conventions

The term "this document" is used to refer to this Recommendation | International Standard.

In this document, the following verbal forms are used:

- "shall" indicates a requirement. When used to express a mandatory constraint on the values of syntax elements or the values of variables derived from these syntax elements, it is the responsibility of the encoder to ensure that the constraint is fulfilled.
- "should" indicates a recommendation. It is used to refer to behaviour of an implementation that is encouraged to be followed under anticipated ordinary circumstances, but is not a requirement for conformance to this document.
- "may" indicates a permission.
- "can" indicates a possibility or a capability.

Information marked as "NOTE" is intended to assist the understanding or use of the document. "Notes to entry" used in [Clause 3](#) provide additional information that supplements the terminological data and can contain provisions relating to the use of a term.

Information technology — MPEG video technologies —

Part 7:

Versatile supplemental enhancement information messages for coded video bitstreams

1 Scope

This document specifies the syntax and semantics of video usability information (VUI) parameters and supplemental enhancement information (SEI) messages. The VUI parameters and SEI messages defined in this document are designed to be conveyed within coded video bitstreams in a manner specified in a video coding specification or to be conveyed by other means determined by the specifications for systems that make use of such coded video bitstreams. This document is particularly intended for use with coded video bitstreams as specified by Rec. ITU-T H.266 | ISO/IEC 23090-3, although it is drafted in a manner intended to be sufficiently generic that it can also be used with other types of coded video bitstreams.

VUI parameters and SEI messages can assist in processes related to decoding, display or other purposes. However, unless otherwise specified in a referencing specification, the interpretation and use of the VUI parameters and SEI messages specified in this document is not a required functionality of a video decoder or receiving video system. Although semantics are specified for the VUI parameters and SEI messages, decoders and receiving video systems can simply ignore the content of the VUI parameters and SEI messages or can use them in a manner that somewhat differs from what is specified in this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10646, *Information technology — Universal coded character set (UCS)*

ISO/IEC 11578:1996, *Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)*

ISO/IEC 15938-17, *Information technology — Multimedia content description interface — Part 17: Compression of neural networks for multimedia content description and analysis*

Rec. ITU-T H.273 | ISO/IEC 23091-2, *Information technology — Coding-independent code points — Part 2: Video*

Rec. ITU-T T.35:2000, *Procedure for the allocation of ITU-T defined codes for non-standard facilities*

ISO/CIE 11664-1, *Colorimetry — Part 1: CIE standard colorimetric observers*

IETF RFC 1321, *The MD5 Message-Digest Algorithm*

IETF RFC 4151, *The 'tag' URI Scheme*

IETF RFC 5646, *Tags for Identifying Languages*.

IETF RFC 3986, *Uniform Resource Identifiers (URI): Generic Syntax*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1

AU

access unit

set of *PUs* that belong to different *layers* and contain *coded pictures* associated with the same *output time*

3.2

APS

adaptation parameter set

syntax structure containing *syntax elements* that apply to zero or more *slices* as determined by zero or more *syntax elements* found in *slice headers*

3.3

alpha blending

process in which an auxiliary *coded picture* is used in combination with a primary *coded picture* and with other data not specified by this document in the display process

Note 1 to entry: In an alpha blending process, the luma samples of an auxiliary coded picture are interpreted as indications of the degree of opacity (or, equivalently, the degrees of transparency) associated with corresponding samples of the primary coded picture.

3.4

associated IRAP picture

previous *IRAP picture* (when present) in *decoding order*, for a particular picture, in the same *layer* as the particular *picture*

3.5

azimuth circle

circle on a sphere connecting all points with the same azimuth value

Note 1 to entry: An azimuth circle is always a *great circle* like a longitude line on the earth.

3.6

byte

sequence of 8 bits, within which, when written or read as a sequence of bit values, the left-most and right-most bits represent the most and least significant bits, respectively, and the bits are written or read from left to right

3.7

chroma

sample array or single sample representing one of the two colour difference signals related to the primary colours, represented by the symbols Cb and Cr

Note 1 to entry: The term chroma is used rather than the term chrominance in order to avoid implying the use of linear light transfer characteristics that is often associated with the term chrominance.

3.8

CLVS

coded layer video sequence

sequence of *PUs* of the same layer that consists, in *decoding order*, of a *CLVSS PU*, followed by zero or more *PUs* that are not *CLVSS PUs*, including all subsequent *PUs* up to but not including any subsequent *PU* that is a *CLVSS PU*

3.9

CLVSS PU

PU in which the *coded picture* is a *CLVSS picture*

3.10

CLVSS

coded layer video sequence start

coded picture that starts a new CLVS as specified in a video coding specification

Note 1 to entry: In Rec. ITU-T H.266 | ISO/IEC 23090-3, a CLVSS picture is an *IRAP picture* with *NoIncorrectPicOutputFlag* equal to 1 or a gradual decoding refresh picture with *NoIncorrectPicOutputFlag* equal to 1. In Rec. ITU-T H.265 | ISO/IEC 23008-2, a CLVSS picture is an *IRAP picture* with *NoRaslOutputFlag* equal to 1.

3.11

coded picture

coded representation of a *picture* containing all *CTUs* of the *picture*

3.12

coded slice NAL unit

NAL unit that contains a *coded slice*

3.13

coded video bitstream

sequence of bits that forms the representation of a sequence of *AUs* forming one or more *CVSSs*

3.14

CVS

coded video sequence

sequence of *AUs* that consists, in *decoding order*, of a *CVSS AU*, followed by zero or more *AUs* that are not *CVSS AUs*, including all subsequent *AUs* up to but not including any subsequent *AU* that is a *CVSS AU*

3.15

CVSS AU

AU that has a *PU* for each *layer* present in the *CVS* and the *coded picture* in each *PU* is a *CLVSS picture*

3.16

component

array or single sample from one of the three arrays (*luma* and two *chroma*) that compose a *picture* in 4:2:0, 4:2:2, or 4:4:4 colour format or the array or a single sample of the array that compose a *picture* in monochrome format

3.17

constituent picture

part of a spatially *frame-packed* stereoscopic *picture* that corresponds to one view, or a *picture* itself when *frame packing* is not in use or the temporal interleaving *frame packing* arrangement is in use

3.18

cropped decoded picture

result of cropping a *decoded picture* based on the conformance cropping window for the corresponding *coded picture*

3.19

decoded picture

decoded picture is derived by decoding a *coded picture*

3.20

decoder

embodiment of a *decoding process*

3.21

decoding order

order in which *syntax elements* are conveyed in the *coded video bitstream* and are processed by a *decoding process*

3.22

decoding process

process that reads a *coded video bitstream* and derives *decoded pictures* from it

3.23

elevation circle

circle on a sphere connecting all points with the same elevation value

Note 1 to entry: An elevation circle is similar to a latitude line on the earth. Except when the elevation value is zero, an elevation circle is not a *great circle* like a longitude circle on the earth.

3.24

encoder

embodiment of an *encoding process*

3.25

encoding process

process that produces a *coded video bitstream*

3.26

field

assembly of alternative rows of samples of a *frame*

3.27

flag

variable or single-bit *syntax element* that can take one of the two possible values: 0 and 1

3.28

frame

composition of a top *field* and a bottom *field*, where sample rows 0, 2, 4, ... originate from the top *field* and sample rows 1, 3, 5, ... originate from the bottom *field*

3.29

global coordinate axes

coordinate axes associated with *omnidirectional video* that are associated with an externally referenceable position and orientation

Note 1 to entry: The global coordinate axes could correspond to the position and orientation of a device or rig used for omnidirectional audio/video acquisition as well as the position of an observer's head in the three-dimensional space of the *omnidirectional video* rendering environment.

3.30

great circle

intersection of a sphere and a plane that passes through the centre point of the sphere

Note 1 to entry: A great circle is also known as an orthodrome or Riemannian circle.

3.31

inter prediction

aspect of the *decoding process* for a *coded picture* that makes use of data derived from the *decoding process* of one or more previously decoded *reference pictures*

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*

3.33**layer**

set of *VCL NAL units* that all have a particular value of layer identifier and the associated non-VCL NAL units, wherein the layer identifier is a variable for which the value is specified by a video coding specification

Note 1 to entry: In the contexts of Rec. ITU-T H.266 | ISO/IEC 23090-3 and Rec. ITU-T H.265 | ISO/IEC 23008-2, the layer identifier is the value of the *nuh_layer_id* syntax element in the NAL unit header.

3.34**leading picture**

picture that is in the same *layer* as the *associated IRAP picture* and precedes the *associated IRAP picture* in *output order*

3.35**local coordinate axes**

coordinate axes having a specified rotation relationship relative to the *global coordinate axes*

3.36**luma**

sample array or single sample representing the monochrome signal related to the primary colours, represented by the symbol or subscript Y or L

Note 1 to entry: The term luma is used rather than the term luminance in order to avoid implying the use of linear light transfer characteristics that is often associated with the term luminance. The symbol L is sometimes used instead of the symbol Y to avoid confusion with the symbol y as used for vertical location.

3.37**NAL unit**

syntax structure containing an indication of the type of data that follows and *bytes* containing that data in a manner that enables the extraction of a string of data bits from the syntax structure

3.38**non-VCL NAL unit**

NAL unit that is not a *VCL NAL unit*

3.39**omnidirectional video**

video content in a format that enables rendering according to the user's viewing orientation, e.g., if viewed using a head-mounted device, or according to a user's desired *viewport*, reflecting a potentially rotated viewing position

3.40**output order**

order in which the *decoded pictures* are output from the *decoder* (for the *decoded pictures* that are to be output from the *decoder*)

3.41**output time**

time when a *decoded picture* is to be output from the *decoder* (for the *decoded pictures* that are to be output from the *decoder*)

3.42**packed region**

region in a *region-wise packed picture* that is mapped to a *projected region* according to a *region-wise packing*

3.43**picture**

array of *luma* samples in monochrome format or an array of *luma* samples and two corresponding arrays of *chroma* samples in 4:2:0, 4:2:2, and 4:4:4 colour format

Note 1 to entry: A picture could be either a frame or a field. However, in one CLVS, either all pictures are frames or all pictures are fields.

3.44**PPS****picture parameter set**

A *syntax structure* containing *syntax elements* that apply to zero or more entire *coded pictures* as determined by a *syntax element* that is the same for all *slices* of a picture and found in the picture header or *slice headers* of each *picture*

3.45**PU****picture unit**

set of *NAL units* that contain all *VCL NAL units* of a *coded picture* and their associated non-VCL *NAL units*

3.46**projected picture**

picture that uses a *projection* format for *omnidirectional video*

3.47**projected region**

region in a *projected picture* that is mapped to a *packed region* according to a *region-wise packing*

3.48**projection**

specified correspondence between the colour samples of a *projected picture* and azimuth and elevation positions on a sphere

3.49**random access**

act of starting the decoding process for a *coded video bitstream* at a point other than the beginning of the bitstream

3.50**RASL picture**

leading picture that cannot be correctly decoded when the decoding process starts from the *associated IRAP picture*

3.51**reference picture**

picture that contains samples that could be used for *inter prediction* in the decoding process of subsequent pictures in decoding order

3.52**reference picture list**

list of *reference pictures* that is used for *inter prediction* of a *slice*

3.53**region-wise packed picture**

decoded picture that contains one or more *packed regions*

Note 1 to entry: A region-wise packed picture could contain a *region-wise packing* of a *projected picture*.

3.54**region-wise packing**

transformation, resizing, and relocation of *packed regions* of a *region-wise packed picture* to remap the *packed regions* to *projected regions* of a *projected picture*

3.55**sample aspect ratio**

indicated width-to-height aspect ratio of the luma samples of the associated *decoded pictures*

3.56**slice**

region of a *picture* that can be decoded separately from other regions of the same *coded picture* (although in some cases the *decoding process* for the *picture* might use *inter prediction* that makes reference to other previously decoded *reference pictures*)

3.57**source**

term used to describe the video material or some of its attributes before encoding

3.58**sphere coordinates**

azimuth and elevation angles identifying a location of a point on a sphere

3.59**sphere region**

region on a sphere, specified either by four *great circles* or by two *azimuth circles* and two *elevation circles*, or such a region on a rotated sphere after applying yaw, pitch, and roll rotations

3.60**STSA picture**

coded picture that enables up-switching, at the *coded picture*, to the *temporal sublayer* containing the *coded picture*, from the immediately lower *temporal sublayer* of the same *layer* when the *coded picture* does not belong to the lowest *temporal sublayer*

Note 1 to entry: An STSA picture does not use pictures in the same layer and with the same temporal sublayer identifier as the STSA picture for inter prediction reference. Pictures following an STSA picture in decoding order in the same layer and with the same temporal sublayer identifier as the STSA picture do not use pictures prior to the STSA picture in decoding order in the same layer and with the same temporal sublayer identifier as the STSA picture for inter prediction reference. STSA pictures in an independent layer (i.e., a layer that does not depend on other layers in its decoding) always have a temporal sublayer identifier greater than 0.

3.61**syntax element**

element of data represented in a *syntax structure*

3.62**syntax structure**

zero or more *syntax elements* that are present together in a specified order in a string of data bits, where the left-most bit is considered to be the first and most significant bit, and the right-most bit is considered to be the last and least significant bit

3.63**SEI message**

syntax structure that provides a particular type of information that assists in processes related to decoding, display or other purposes but is not needed by the *decoding process* in order to determine the values of the samples in *decoded pictures*

3.64**temporal sublayer**

subset of a temporal scalable *bitstream*, consisting of *VCL NAL units* with a particular value of *temporal sublayer identifier* and the associated *non-VCL NAL units*

3.65**temporal sublayer identifier**

number greater than or equal to 0 defined by a variable for which the value is specified by a video coding specification such that pictures of all *temporal sublayers* have a specified temporal output order relative to each other and pictures with a lower temporal sublayer identifier can be decoded without reference to pictures with a higher temporal sublayer identifier

3.66**tilt angle**

angle indicating the amount of tilt of a *sphere region*, measured as the amount of rotation of a *sphere region* along the axis originating from the sphere origin passing through the centre point of the *sphere region*, where the angle value increases clockwise when looking from the origin towards the positive end of the axis

3.67**trailing picture**

coded picture that is associated with an *IRAP picture* and follows the *IRAP picture* in both decoding order and output order

3.68**VCL NAL unit**

collective term for *coded slice NAL units* and the subset of other *NAL units* that have *reserved* values of *NAL unit* type identifiers that are classified as VCL NAL units in a referencing specification

3.69**VUI parameters**

syntax structure that identifies properties of interpretation of decoded pictures for display purposes, particularly including colour representation information

3.70**viewport**

region of *omnidirectional video* content suitable for display and viewing by the user

4 Abbreviated terms

ACI	alpha channel information
AU	access unit
APS	adaptation parameter set
CLVS	coded layer video sequence
CLVSS	coded layer video sequence start
CRC	cyclic redundancy check
CTI	colour transform information
CVS	coded video sequence
DRAP	dependent random access point
DRI	depth representation information
EDRAP	extended dependent random access point
FIR	finite impulse response
IRAP	intra random access point
MAI	multiview acquisition information
NAL	network abstraction layer
NNPF	neural-network post-processing filter
NNPFA	neural-network post-filter activation

NNPFC	neural-network post-filter characteristics
PPS	picture parameter set
PU	picture unit
RASL	random access skipped leading
RWP	region-wise packing
SAR	sample aspect ratio
SARI	sample aspect ratio information
SDI	scalability dimension information
SEI	supplemental enhancement information
STSA	step-wise temporal sublayer access
URI	uniform resource identifier
VCL	video coding layer
VUI	video usability information

5 Conventions

5.1 General

The word "reserved" is used to specify that some values of a particular syntax element are for future use by ITU-T | ISO/IEC and shall not be used in syntax structures conforming to this document, but could potentially be used in syntax structures conforming to future editions of this document by ITU-T | ISO/IEC.

The word "unspecified" is used to describe some values of a particular syntax element to indicate that the values have no specified meaning in this document and are not expected to have a specified meaning in future editions of this document by ITU-T | ISO/IEC.

The mathematical operators used in this document are similar to those used in the C programming language. However, the results of integer division and arithmetic shift operations are defined more precisely, and additional operations are defined, such as exponentiation and real-valued division.

Numbering and counting conventions generally begin from 0 (e.g. "the first" is equivalent to the 0-th, "the second" is equivalent to the 1-th, etc.).

5.2 Arithmetic operators

+	addition
–	subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	multiplication, including matrix multiplication
x^y	exponentiation Specifies x to the power of y . In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.
/	integer division with truncation of the result toward zero For example, $7 / 4$ and $-7 / -4$ are truncated to 1 and $-7 / 4$ and $7 / -4$ are truncated to -1.

$\div$	division in mathematical equations where no truncation or rounding is intended
$\frac{x}{y}$	division in mathematical equations where no truncation or rounding is intended
$\sum_{i=x}^y f(i)$	summation of $f(i)$ with i taking all integer values from x up to and including y
$x \% y$	modulus Remainder of x divided by y , defined only for integers x and y with $x \geq 0$ and $y > 0$

5.3 Logical operators

$x \&\& y$	Boolean logical "and" of x and y
$x y$	Boolean logical "or" of x and y
$!$	Boolean logical "not"
$x ? y : z$	if x is TRUE or not equal to 0, evaluates to the value of y ; otherwise, evaluates to the value of z

5.4 Relational operators

$>$	greater than
$>=$	greater than or equal to
$<$	less than
$<=$	less than or equal to
$=$	equal to
$!=$	not equal to

When a relational operator is applied to a syntax element or variable that has been assigned the value "na" (not applicable), the value "na" is treated as a distinct value for the syntax element or variable. The value "na" is considered not to be equal to any other value.

5.5 Bit-wise operators

$\&$	bit-wise "and" When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
$ $	bit-wise "or" When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
$\wedge$	bit-wise "exclusive or" When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

- $x \gg y$ arithmetic right shift of a two's complement integer representation of x by y binary digits
This function is defined only for non-negative integer values of y . Bits shifted into the most significant bits (MSBs) as a result of the right shift have a value equal to the MSB of x prior to the shift operation.
- $x \ll y$ arithmetic left shift of a two's complement integer representation of x by y binary digits
This function is defined only for non-negative integer values of y . Bits shifted into the least significant bits (LSBs) as a result of the left shift have a value equal to 0.

5.6 Assignment operators

- $=$ assignment operator
- $++$ increment, i.e., $x++$ is equivalent to $x = x + 1$; when used in an array index, evaluates to the value of the variable prior to the increment operation
- $--$ decrement, i.e., $x--$ is equivalent to $x = x - 1$; when used in an array index, evaluates to the value of the variable prior to the decrement operation
- $+=$ increment by amount specified, i.e., $x += 3$ is equivalent to $x = x + 3$, and $x += (-3)$ is equivalent to $x = x + (-3)$
- $-=$ decrement by amount specified, i.e., $x -= 3$ is equivalent to $x = x - 3$, and $x -= (-3)$ is equivalent to $x = x - (-3)$

5.7 Range notation

- $x = y..z$ x takes on integer values starting from y to z , inclusive, with x , y , and z being integer numbers and z being greater than y

5.8 Mathematical functions

$$\text{Abs}(x) = \begin{cases} x & ; \ x \geq 0 \\ -x & ; \ x < 0 \end{cases} \quad (1)$$

$\text{Asin}(x)$ trigonometric inverse sine function, operating on an argument x that is in the range of -1.0 to 1.0 , inclusive, with an output value in the range of $-\pi/2$ to $\pi/2$, inclusive, in units of radians (2)

$\text{Atan}(x)$ trigonometric inverse tangent function, operating on an argument x , with an output value in the range of $-\pi/2$ to $\pi/2$, inclusive, in units of radians (3)

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x = 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{otherwise} \end{cases} \quad (4)$$

$\text{Ceil}(x)$ smallest integer greater than or equal to x (5)

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases} \quad (6)$$

$\text{Cos}(x)$ trigonometric cosine function operating on an argument x in units of radians (7)

$\text{Floor}(x)$ largest integer less than or equal to x (8)

$\text{Ln}(x)$ natural logarithm of x (the base- e logarithm, where e is the natural logarithm base constant 2.718 281 828...) (9)

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases} \quad (10)$$

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases} \quad (11)$$

$$\text{Reflect}(x, y) = \begin{cases} \text{Min}(-y, x) & ; \quad y < 0 \\ \text{Max}(x - (y - x), 0) & ; \quad y > x \\ y & ; \quad \text{otherwise} \end{cases} \quad (12)$$

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5) \quad (13)$$

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x = 0 \\ -1 & ; \quad x < 0 \end{cases} \quad (14)$$

$\text{Sin}(x)$ trigonometric sine function operating on an argument x in units of radians (15)

$\text{Sqrt}(x)$ square root of x (16)

$\text{Tan}(x)$ trigonometric tangent function operating on an argument x in units of radians (17)

$$\text{Wrap}(x, y) = \begin{cases} \text{Max}(0, x + y + 1) & ; \quad y < 0 \\ \text{Min}(x, y - x - 1) & ; \quad y > x \\ y & ; \quad \text{otherwise} \end{cases} \quad (18)$$

5.9 Order of operation precedence

When order of precedence in an expression is not indicated explicitly by use of parentheses, the following rules apply:

- Operations of a higher precedence are evaluated before any operation of a lower precedence.
- Operations of the same precedence are evaluated sequentially from left to right.

[Table 1](#) specifies the precedence of operations from highest to lowest; a higher position in the table indicates a higher precedence.

NOTE For those operators that are also used in the C programming language, the order of precedence used in this document is the same as used in the C programming language.

Table 1 — Operation precedence from highest (at top of table) to lowest (at bottom of table)

Operations (with operands x, y, and z)
"x++", "x--"
"!x", "-x" (as a unary prefix operator)
x^y
"x * y", "x / y", "x ÷ y", " $\frac{x}{y}$ ", "x % y"
"x + y", "x - y" (as a two-argument operator), " $\sum_{i=x}^y f(i)$ "
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x = y", "x != y"
"x & y"
"x y"
"x && y"
"x y"
"x ? y : z"
"x.y"
"x = y", "x += y", "x -= y"

5.10 Variables, syntax elements and tables

Syntax elements in the syntax tables are represented in **bold** type. Each syntax element is described by its name (all lower case letters with underscore characters), and one descriptor for its method of coded representation. The decoding process behaves according to the value of the syntax element and to the values of previously decoded syntax elements. When a value of a syntax element is used in the syntax tables or the text, it appears in regular (i.e., not bold) type.

In some cases the syntax tables and semantics use the values of other variables derived from the values of syntax elements. Such variables appear in the syntax tables, or text, named by a mixture of lower case and upper case letter and without any underscore characters. Variables starting with an upper case letter are derived for the decoding of the current syntax structure and all depending syntax structures. Variables starting with an upper case letter could, in some cases, be used in the decoding process for later syntax

structures without mentioning the originating syntax structure of the variable. Variables starting with a lower case letter are only used within the subclause in which they are derived.

In some cases, "mnemonic" names for syntax element values or variable values are used interchangeably with their numerical values. Sometimes "mnemonic" names are used without any associated numerical values. The association of values and names is specified in the text. The names are constructed from one or more groups of letters separated by an underscore character. Each group starts with an upper case letter and could contain more upper case letters.

NOTE The syntax is described in a manner that closely follows the C-language syntactic constructs.

Functions that specify properties of the current position in the SEI message payload data are referred to as syntax functions. These functions are specified in [subclause 6.3](#) and assume the existence of a pointer with an indication of the position of the next bit to be read by the decoding process from the payload data. Syntax functions are described by their names, which are constructed as syntax element names and end with left and right round parentheses including zero or more variable names (for definition) or values (for usage), separated by commas (if more than one variable).

Functions that are not syntax functions (including mathematical functions specified in [subclause 5.8](#)) are described by their names, which start with an upper case letter, contain a mixture of lower and upper case letters without any underscore character, and end with left and right parentheses including zero or more variable names (for definition) or values (for usage) separated by commas (if more than one variable).

A one-dimensional array is referred to as a list. A two-dimensional array is referred to as a matrix. Arrays can either be syntax elements or variables. Subscripts or square parentheses are used for the indexing of arrays. In reference to a visual depiction of a matrix, the first subscript is used as a row (vertical) index and the second subscript is used as a column (horizontal) index. The indexing order is reversed when using square parentheses rather than subscripts for indexing. Thus, an element of a matrix s at horizontal position x and vertical position y could be denoted either as $s[x][y]$ or as s_{yx} . A single column of a matrix could be referred to as a list and denoted by omission of the row index. Thus, the column of a matrix s at horizontal position x could be referred to as the list $s[x]$.

A specification of values of the entries in rows and columns of an array could be denoted by $\{ \{ \dots \} \{ \dots \} \}$, where each inner pair of brackets specifies the values of the elements within a row in increasing column order and the rows are ordered in increasing row order. Thus, setting a matrix s equal to $\{ \{ 1 \ 6 \} \{ 4 \ 9 \} \}$ specifies that $s[0][0]$ is set equal to 1, $s[1][0]$ is set equal to 6, $s[0][1]$ is set equal to 4, and $s[1][1]$ is set equal to 9.

Binary notation is indicated by enclosing the string of bit values by single quote marks. For example, '01000001' represents an eight-bit string having only its second and its last bits (counted from the most to the least significant bit) equal to 1.

Hexadecimal notation, indicated by prefixing the hexadecimal number by "0x", is used in some cases instead of binary notation when the number of bits is an integer multiple of 4. For example, 0x41 represents an eight-bit string having only its second and its last bits (counted from the most to the least significant bit) equal to 1.

Numerical values not enclosed in single quotes and not prefixed by "0x" are decimal values.

A value equal to 0 represents a FALSE condition in a test statement. The value TRUE is represented by any value different from zero.

5.11 Text description of logical operations

In the text, a statement of logical operations as would be described mathematically in the following form:

```

if( condition 0 )
    statement 0
else if( condition 1 )
    statement 1
...
else /* informative remark on remaining condition */
    statement n

```

is typically described in the following manner:

... as follows / ... the following applies:

- If condition 0, statement 0
- Otherwise, if condition 1, statement 1
- ...
- Otherwise (informative remark on remaining condition), statement n

Each "If ... Otherwise, if ... Otherwise, ..." statement in the text is introduced with "... as follows" or "... the following applies" immediately followed by "If ... ". The last condition of the "If ... Otherwise, if ... Otherwise, ..." is always an "Otherwise, ...". Interleaved "If ... Otherwise, if ... Otherwise, ..." statements can be identified by matching "... as follows" or "... the following applies" with the ending "Otherwise, ...".

In the text, a statement of logical operations as would be described mathematically in the following form:

```

if( condition 0a && condition 0b )
    statement 0
else if( condition 1a || condition 1b )
    statement 1
...
else
    statement n

```

is typically described in the following manner:

... as follows / ... the following applies:

- If all of the following conditions are true, statement 0:
 - condition 0a
 - condition 0b
- Otherwise, if one or more of the following conditions are true, statement 1:
 - condition 1a
 - condition 1b
- ...
- Otherwise, statement n

In the text, a statement of logical operations as would be described mathematically in the following form:

```

if( condition 0 )
    statement 0
if( condition 1 )
    statement 1

```

is typically described in the following manner:

When condition 0, statement 0

When condition 1, statement 1

5.12 Processes

Processes are used to describe the decoding of syntax elements. A process has a separate specification and invoking. All syntax elements and upper case variables that pertain to the current syntax structure and depending syntax structures are available in the process specification and invoking. A process specification might also have a lower case variable explicitly specified as input. Each process specification has explicitly specified an output. The output is a variable that can either be an upper case variable or a lower case variable.

When invoking a process, the assignment of variables is specified as follows:

- If the variables at the invoking and the process specification do not have the same name, the variables are explicitly assigned to lower case input or output variables of the process specification.
- Otherwise (the variables at the invoking and the process specification have the same name), assignment is implied.

6 Syntax and semantics

6.1 General

It is intended that this document is referenced by other technical specifications, which should specify certain necessary elements to enable the use of the specified VUI parameters and SEI messages.

Technical specifications that reference this document for carrying VUI parameters syntax structure shall specify a container to carry the data of the VUI parameters syntax structure and to identify the length in bits of the VUI parameters syntax structure, e.g., the `vui_payload( )` syntax structure specified in Rec. ITU-T H.266 | ISO/IEC 23090-3. The design of the container should provide the ability to detect the number of bits in the `vui_parameters( )` syntax structure and to allow the number of bits to be increased in future editions of this document, thus enabling this document to provide extensibility by directly appending additional syntax elements to the end of the `vui_parameters( )` syntax structure in future editions of this document. The syntax of the container of the `vui_parameters( )` syntax structure is outside the scope of this document.

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

The length of the VUI parameters syntax structure or an SEI message 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 $\text{Ceil}(\text{PayloadBits} \div 8)$.

For the VUI parameters and most of the SEI messages specified in this edition of this document (other than the filler payload, user data registered, user data unregistered, and reserved SEI messages), the values of `PayloadBits` and `payloadSize` are not used for the parsing of the syntax. However, in some future edition of this document, the value of `PayloadBits` or `payloadSize` could be used as part of the syntax specification for these syntax structures, for example to identify whether payload extension data is present in the

VUI parameters or in an SEI message syntax structure that was not specified in an earlier edition of this document.

The syntax specification in Rec. ITU-T H.266 | ISO/IEC 23090-3 and Rec. ITU-T H.265 | ISO/IEC 23008-2 establishes, under some circumstances, a certain pattern of bits that is used for detecting the value of PayloadBits. It is expected that future editions of this document will be written to ensure that such future editions will be compatible with the pattern for extension data that is specified in those other specifications. This pattern is such that when extension data is present and the last bit of such extension data is the last (least significant) bit of a byte, the extension data ends with a byte that contains a bit equal to 1 followed by 7 bits that are equal to 0.

It is a requirement of bitstream conformance to this edition of this document that the value of PayloadBits, as determined by this external means, shall be equal to the number of bits in the VUI parameters syntax structure or the SEI message syntax structure, as applicable.

It is a requirement of decoder conformance to this edition of this document that when PayloadBits is greater than the number of bits in the VUI parameters syntax structure or an SEI message syntax structure, the extra data at the end of the VUI or SEI payload data shall be ignored. The semantics for such extra data could potentially be specified in some future edition of this document.

For example, each SEI message could be carried as a string of data bits that is prefixed with an SEI message payload type indication derived as a payloadType variable within a NAL unit that could contain emulation prevention bytes as specified in Rec. ITU-T H.266 | ISO/IEC 23090-3. When such emulation prevention bytes are present, the emulation prevention bytes are not counted when determining the values of PayloadBits and payloadSize.

6.2 Method of specifying syntax in tabular form

The syntax tables in this document specify a superset of the syntax of the VUI parameters and all allowed SEI messages. Additional constraints on the syntax are specified, either directly or indirectly, in other subclauses.

This subclause lists examples of the syntax specification format. When **syntax_element** appears, it specifies that a syntax element is parsed from the VUI parameters syntax or an SEI message syntax and the data pointer is advanced to the next position beyond the syntax element in the syntax parsing process.

	Descriptor
/* A statement can be a syntax element with an associated descriptor or can be an expression used to specify conditions for the existence, type and quantity of syntax elements, as in the following two examples */	
syntax_element	ue(v)
conditioning statement	
/* A group of statements enclosed in curly brackets is a compound statement and is treated functionally as a single statement. */	
{	
statement	
statement	
...	
}	
/* A "while" structure specifies a test of whether a condition is true, and if true, specifies evaluation of a statement (or compound statement) repeatedly until the condition is no longer true */	
while(condition)	

statement	
/* A "do ... while" structure specifies evaluation of a statement once, followed by a test of whether a condition is true, and if true, specifies repeated evaluation of the statement until the condition is no longer true */	
do	
statement	
while(condition)	
/* An "if ... else" structure specifies a test of whether a condition is true and, if the condition is true, specifies evaluation of a primary statement, otherwise, specifies evaluation of an alternative statement. The "else" part of the structure and the associated alternative statement is omitted if no alternative statement evaluation is needed */	
if(condition)	
primary statement	
else	
alternative statement	
/* A "for" structure specifies evaluation of an initial statement, followed by a test of a condition, and if the condition is true, specifies repeated evaluation of a primary statement followed by a subsequent statement until the condition is no longer true. */	
for(initial statement; condition; subsequent statement)	
primary statement	

6.3 Specification of syntax functions and descriptors

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

more_data_in_payload() is specified as follows:

- If byte_aligned() is equal to TRUE and the current position in an SEI message syntax structure or vui_parameters() syntax structure is $8 * \text{payloadSize}$ bits from the beginning of the syntax structure, the return value of more_data_in_payload() is equal to FALSE.
- Otherwise, the return value of more_data_in_payload() is equal to TRUE.

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

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

- b(8): byte having any pattern of bit string (8 bits). The parsing process for this descriptor is specified by the return value of the function read_bits(8).
- f(n): fixed-pattern bit string using n bits written (from left to right) with the left bit first. The parsing process for this descriptor is specified by the return value of the function read_bits(n).
- i(n): signed integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function read_bits(n) interpreted as a two's complement integer representation with most significant bit written first.

- **se(v)**: signed integer 0-th order Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in [Clause 9](#) with the order k equal to 0.
 - **st(v)**: null-terminated string encoded as universal coded character set (UCS) transmission format-8 (UTF-8) characters as specified in ISO/IEC 10646. The parsing process is specified as follows: **st(v)** begins at a byte-aligned position in the bitstream and reads and returns a series of bytes from the bitstream, beginning at the current position and continuing up to but not including the next byte-aligned byte that is equal to 0x00, and advances the bitstream pointer by $(\text{stringLength} + 1) * 8$ bit positions, where **stringLength** is equal to the number of bytes returned.
- NOTE The **st(v)** syntax descriptor is only used in this document when the current position in the bitstream is a byte-aligned position.
- **u(n)**: unsigned integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function `read_bits( n )` interpreted as a binary representation of an unsigned integer with most significant bit written first.
 - **ue(v)**: unsigned integer 0-th order Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in [Clause 9](#) with the order k equal to 0.

7 Video usability information parameters

7.1 General

This clause specifies the syntax and semantics for VUI parameters.

When any information regarding the interpretation of the pictures is not present in the `vui_parameters( )` syntax structure, or the `vui_parameters( )` syntax structure is not present, there may be some external means that controls the interpretation.

7.2 VUI parameters syntax

<code>vui_parameters(payloadSize) {</code>	Descriptor
<code>vui_progressive_source_flag</code>	<code>u(1)</code>
<code>vui_interlaced_source_flag</code>	<code>u(1)</code>
<code>vui_non_packed_constraint_flag</code>	<code>u(1)</code>
<code>vui_non_projected_constraint_flag</code>	<code>u(1)</code>
<code>vui_aspect_ratio_info_present_flag</code>	<code>u(1)</code>
<code>if(vui_aspect_ratio_info_present_flag) {</code>	
<code>vui_aspect_ratio_constant_flag</code>	<code>u(1)</code>
<code>vui_aspect_ratio_idc</code>	<code>u(8)</code>
<code>if(vui_aspect_ratio_idc == 255) {</code>	
<code>vui_sar_width</code>	<code>u(16)</code>
<code>vui_sar_height</code>	<code>u(16)</code>
<code>}</code>	
<code>}</code>	
<code>vui_overscan_info_present_flag</code>	<code>u(1)</code>
<code>if(vui_overscan_info_present_flag)</code>	
<code>vui_overscan_appropriate_flag</code>	<code>u(1)</code>
<code>vui_colour_description_present_flag</code>	<code>u(1)</code>
<code>if(vui_colour_description_present_flag) {</code>	

vui_colour primaries	u(8)
vui_transfer_characteristics	u(8)
vui_matrix_coeffs	u(8)
vui_full_range_flag	u(1)
}	
vui_chroma_loc_info_present_flag	u(1)
if(vui_chroma_loc_info_present_flag) {	
if(vui_progressive_source_flag && !vui_interlaced_source_flag)	
vui_chroma_sample_loc_type_frame	ue(v)
else {	
vui_chroma_sample_loc_type_top_field	ue(v)
vui_chroma_sample_loc_type_bottom_field	ue(v)
}	
}	
}	

7.3 VUI parameters semantics

VUI parameters apply to one or more CLVSs.

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

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

- A chroma format indicator, denoted herein by *ChromaFormatIdc*, such that the value 0 indicates that the picture has only a luma component and other values indicate that the picture has three colour components that consist of a luma component and two associated chroma components, such that the width and height of each chroma component are the width and height of the luma component divided by *SubWidthC* and *SubHeightC*, respectively, where *SubWidthC* and *SubHeightC* are determined from *ChromaFormatIdc* as specified by [Table 2](#).
- A bit depth for the samples of the luma component, denoted herein by *BitDepth_Y*, and when *ChromaFormatIdc* is not equal to 0, a bit depth for the samples of the two associated chroma components, denoted herein by *BitDepth_C*.

Table 2 — SubWidthC and SubHeightC values derived from ChromaFormatIdc

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

vui_progressive_source_flag and **vui_interlaced_source_flag** are interpreted as follows:

- If *vui_progressive_source_flag* is equal to 1 and *vui_interlaced_source_flag* is equal to 0, the source scan type of the pictures should be interpreted as progressive only.
- Otherwise, if *vui_progressive_source_flag* is equal to 0 and *vui_interlaced_source_flag* is equal to 1, the source scan type of the pictures should be interpreted as interlaced only.

- Otherwise, if `vui_progressive_source_flag` is equal to 0 and `vui_interlaced_source_flag` is equal to 0, the source scan type of the pictures should be interpreted as unknown or unspecified or specified by external means not specified in this document.
- Otherwise (`vui_progressive_source_flag` is equal to 1 and `vui_interlaced_source_flag` is equal to 1), the source scan type of each picture is indicated at the picture level using the syntax element `ffi_source_scan_type` in a frame-field information SEI message.

`vui_non_packed_constraint_flag` equal to 1 specifies that there shall not be any frame packing arrangement SEI messages present in the bitstream that apply to the CLVS. `vui_non_packed_constraint_flag` equal to 0 does not impose such a constraint.

`vui_non_projected_constraint_flag` equal to 1 specifies that there shall not be any equirectangular projection SEI messages or generalized cubemap projection SEI messages present in the bitstream that apply to the CLVS. `vui_non_projected_constraint_flag` equal to 0 does not impose such a constraint.

`vui_aspect_ratio_info_present_flag` equal to 1 specifies that `vui_aspect_ratio_idc` is present. `vui_aspect_ratio_info_present_flag` equal to 0 specifies that `vui_aspect_ratio_idc` is not present.

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

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

`vui_sar_width`, when present, indicates the horizontal size of the SAR (in arbitrary units) of the luma samples of decoded pictures in the CLVS, unless indicated otherwise by associated SARI SEI messages when `vui_aspect_ratio_constant_flag` is equal to 0.

`vui_sar_height`, when present, indicates the vertical size of the SAR (in the same arbitrary units as `vui_sar_width`) of the luma samples of decoded pictures in the CLVS, unless indicated otherwise by associated SARI SEI messages when `vui_aspect_ratio_constant_flag` is equal to 0.

When present, `vui_sar_width` and `vui_sar_height` shall be relatively prime or equal to 0. When `vui_aspect_ratio_idc` is equal to 0 or `vui_sar_width` is equal to 0 or `vui_sar_height` is equal to 0, the SAR is unknown or unspecified in this document or may be determined by other means, such as the SARI SEI message.

`vui_overscan_info_present_flag` equal to 1 specifies that the `vui_overscan_appropriate_flag` is present. When `vui_overscan_info_present_flag` is equal to 0 or is not present, the preferred display method for the video signal is unknown or unspecified or specified by external means.

`vui_overscan_appropriate_flag` equal to 1 indicates that the cropped decoded pictures output are suitable for display using overscan. `vui_overscan_appropriate_flag` equal to 0 indicates that the cropped decoded pictures output contain visually important information in the entire region out to the edges of the conformance cropping window of the picture, such that the cropped decoded pictures output should not be displayed using overscan. Instead, they should be displayed using either an exact match between the display area and the conformance cropping window, or using underscan. As used in this paragraph, the term "overscan" refers to display processes in which some parts near the borders of the cropped decoded pictures are not visible in the display area. The term "underscan" describes display processes in which the entire cropped decoded pictures are visible in the display area, but they do not cover the entire display area.

For display processes that neither use overscan nor underscan, the display area exactly matches the area of the cropped decoded pictures.

NOTE 2 For example, `vui_overscan_appropriate_flag` equal to 1 could be used for entertainment television programming or for a live view of people in a videoconference, and `vui_overscan_appropriate_flag` equal to 0 could be used for computer screen capture or security camera content.

`vui_colour_description_present_flag` equal to 1 specifies that `vui_colour_primaries`, `vui_transfer_characteristics`, and `vui_matrix_coeffs` are present. `vui_colour_description_present_flag` equal to 0 specifies that `vui_colour_primaries`, `vui_transfer_characteristics`, and `vui_matrix_coeffs` are not present.

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

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

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

`vui_matrix_coeffs` shall not be equal to 0 unless both of the following conditions are true:

- `BitDepthC` is equal to `BitDepthY`.
- `ChromaFormatIdc` is equal to 3 (the 4:4:4 chroma format).

The specification of the use of `vui_matrix_coeffs` equal to 0 under all other conditions is reserved for future use by ITU-T | ISO/IEC.

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

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

The specification of the use of `vui_matrix_coeffs` equal to 8 under all other conditions is reserved for future use by ITU-T | ISO/IEC.

When the `vui_matrix_coeffs` syntax element is not present, the value of `vui_matrix_coeffs` is inferred to be equal to 2 (unknown or unspecified or determined by other means not specified in this document).

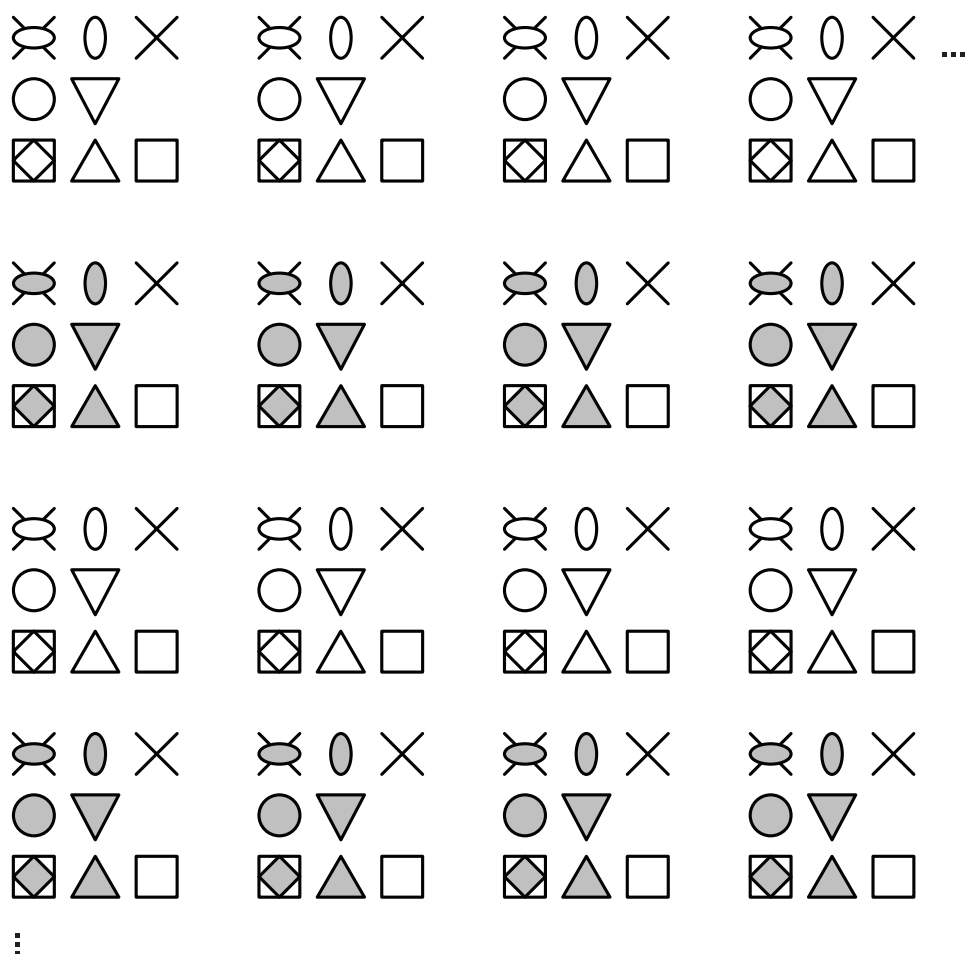
`vui_full_range_flag` indicates the scaling and offset values applied in association with the matrix coefficients. Its semantics are as specified for the `VideoFullRangeFlag` parameter in Rec. ITU-T H.273 | ISO/IEC 23091-2. When not present, the value of `vui_full_range_flag` is inferred to be equal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

`vui_chroma_loc_info_present_flag` equal to 1 specifies that either `vui_chroma_sample_loc_type_frame` or both `vui_chroma_sample_loc_type_top_field` and `vui_chroma_sample_loc_type_bottom_field` are present. `vui_chroma_loc_info_present_flag` equal to 0 specifies that `vui_chroma_sample_loc_type_frame`, `vui_chroma_sample_loc_type_top_field`, and `vui_chroma_sample_loc_type_bottom_field` are not present.

When `ChromaFormatIdc` is not equal to 1, `vui_chroma_loc_info_present_flag` should be equal to 0.

vui_chroma_sample_loc_type_frame, **vui_chroma_sample_loc_type_top_field**, and **vui_chroma_sample_loc_type_bottom_field**, when present, specify the location of chroma samples as follows:

- If GeneralProgressiveSourceFlag is equal to 1, GeneralInterlacedSourceFlag is equal to 0, and ChromaFormatIdc is equal to 1 (4:2:0 chroma format), vui_chroma_sample_loc_type_frame specifies the location of chroma samples for both fields of each frame of the CLVS as shown in [Figure 1](#).
- Otherwise, if ChromaFormatIdc is equal to 1 (4:2:0 chroma format), vui_chroma_sample_loc_type_top_field and vui_chroma_sample_loc_type_bottom_field specify the location of chroma samples for each top field and bottom field of the CLVS, respectively, as shown in [Figure 1](#).
- Otherwise (ChromaFormatIdc is not equal to 1), the values of the syntax elements chroma_sample_loc_type, vui_chroma_sample_loc_type_top_field and vui_chroma_sample_loc_type_bottom_field shall be ignored.



Key

Luma samples:



Chroma samples, where filled symbols indicate a bottom field sample type and empty symbols indicate a top field sample type:



Figure 1 — Location of chroma samples for top and bottom fields for ChromaFormatIdc equal to 1 (4:2:0 chroma format) as a function of vui_chroma_sample_loc_type_top_field and vui_chroma_sample_loc_type_bottom_field in the range of 0 to 5, inclusive

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

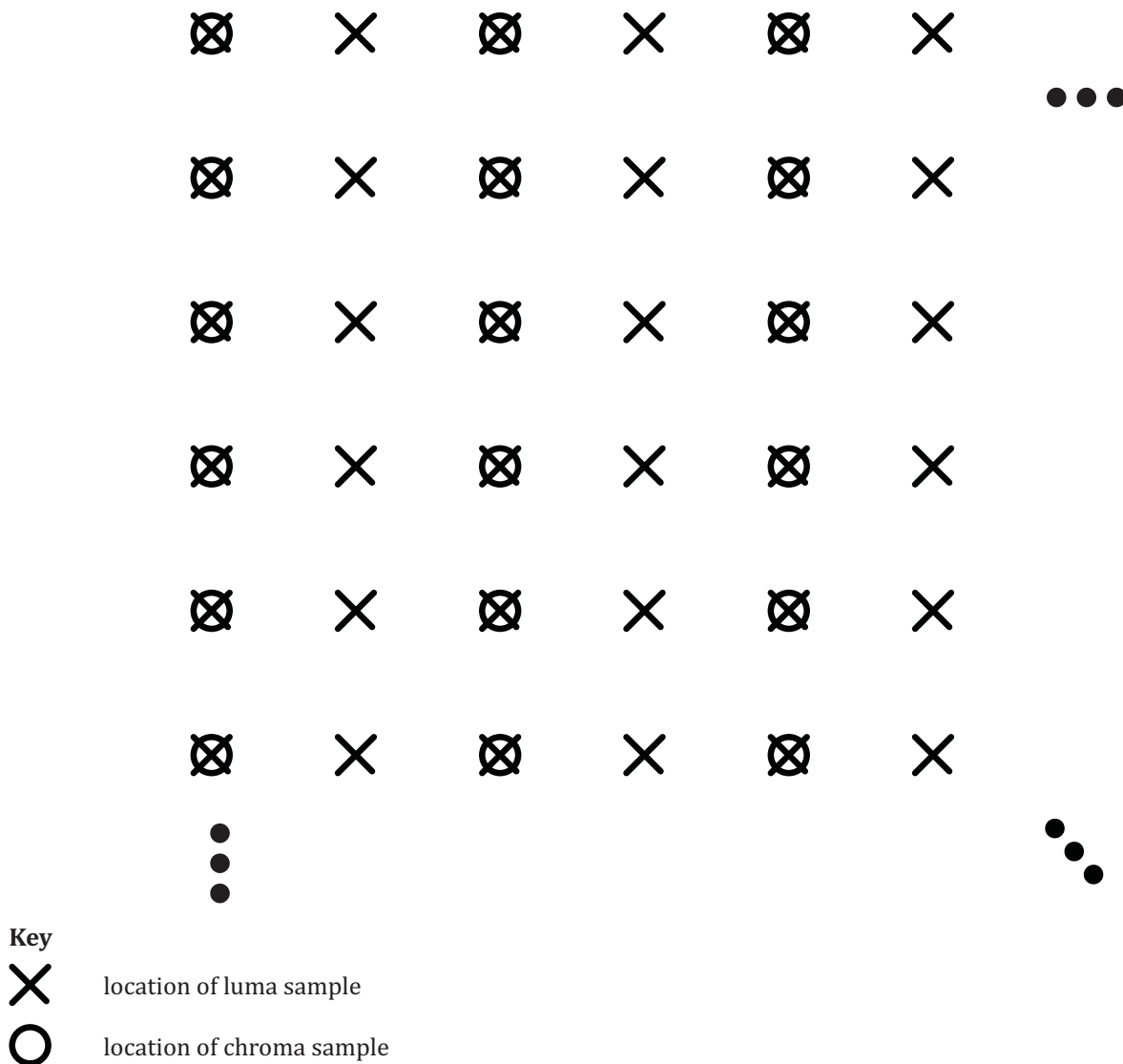


Figure 2 — Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a picture

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

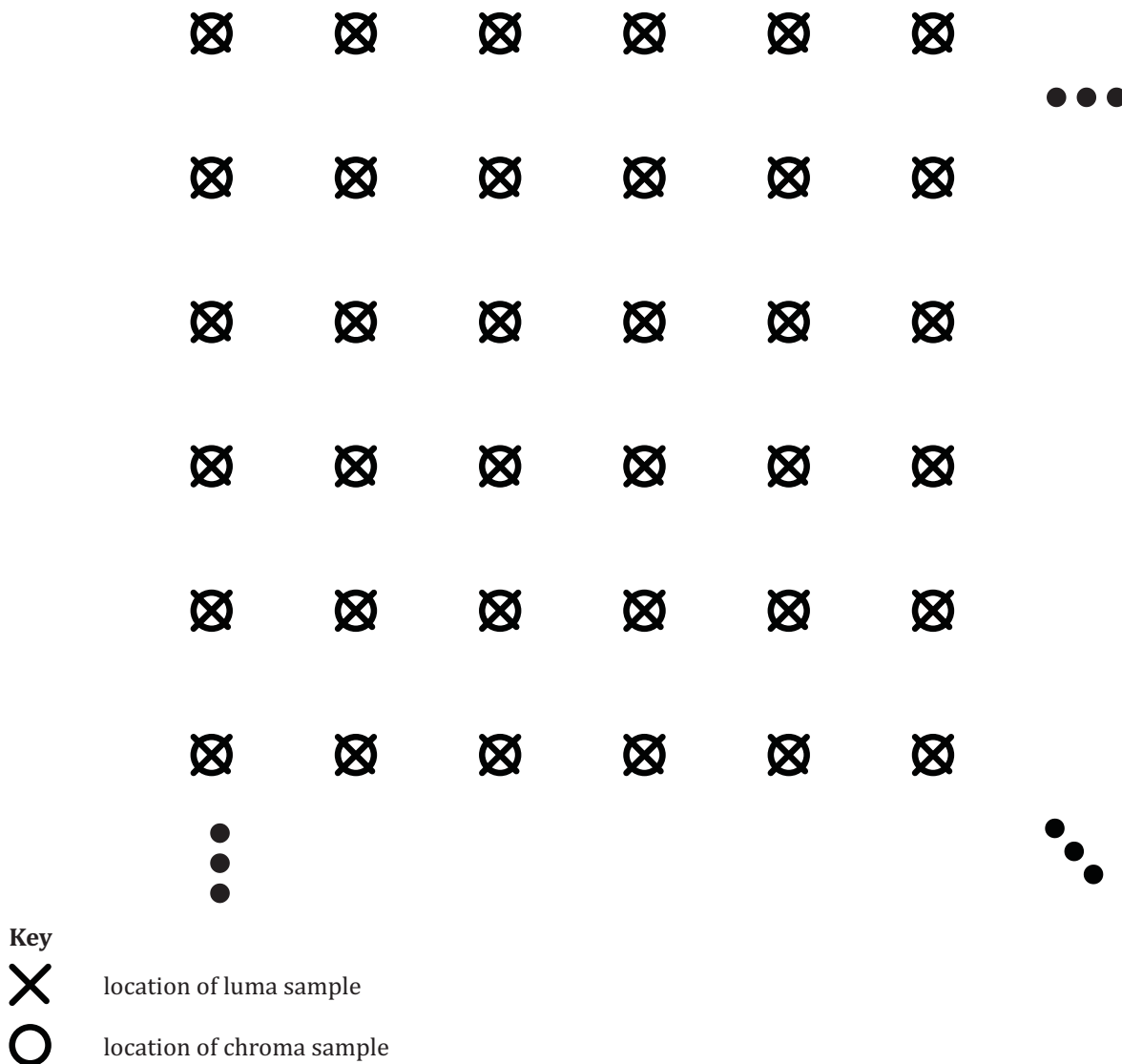


Figure 3 — Nominal vertical and horizontal locations of 4:4:4 luma and chroma samples in a picture

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

When present, the values of vui_chroma_sample_loc_type_frame, vui_chroma_sample_loc_type_top_field and vui_chroma_sample_loc_type_bottom_field shall be in the range of 0 to 6, inclusive.

When ChromaFormatIdc is equal to 1 and vui_chroma_loc_info_present_flag is equal to 0, vui_chroma_sample_loc_type_frame is not present and is inferred to be equal to 6, which indicates that the the location of the chroma samples is unknown or unspecified or specified by other means not specified in this document. When vui_chroma_sample_loc_type_top_field and vui_chroma_sample_loc_type_bottom_field are not present, the values of vui_chroma_sample_loc_type_top_field and vui_chroma_sample_loc_type_bottom_field are inferred to be equal to vui_chroma_sample_loc_type_frame.

NOTE 3 In Rec. ITU-T H.266 | ISO/IEC 23090-3 and Rec. ITU-T H.265 | ISO/IEC 23008-2, a nominal chroma sampling type is identified for ChromaFormatIdc equal to 1 that corresponds to vui_chroma_sample_loc_type_frame, vui_chroma_sample_loc_type_top_field and vui_chroma_sample_loc_type_bottom_field equal to 0.

[Figure 4](#) illustrates the indicated relative position of the top-left chroma sample when ChromaFormatIdc is equal to 1 (i.e., the 4:2:0 chroma format), and vui_chroma_sample_loc_type_top_field or vui_chroma_sample_loc_type_bottom_field is equal to the value of a variable Chroma420LocType. The region represented by the top-left 4:2:0 chroma sample (depicted as a large grey, solid-line square with a large grey dot at its centre) is shown relative to the region represented by the top-left luma sample (depicted as a small black square with

a small black dot at its centre). The regions represented by neighbouring luma samples are depicted as small grey, dotted-line squares with small grey dots at their centres.

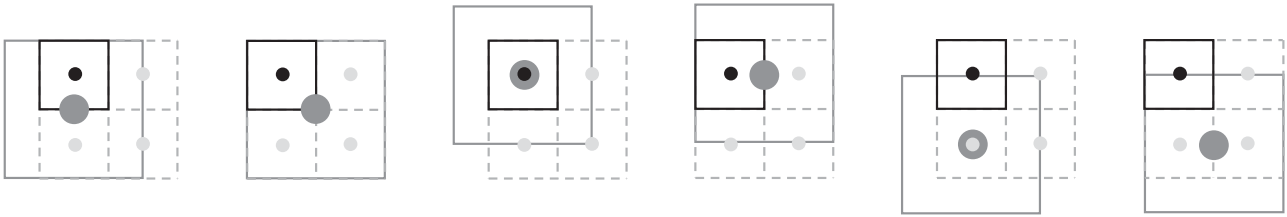


Figure 4 — Location of the top-left chroma sample when ChromaFormatIdc is equal to 1 (4:2:0 chroma format) and Chroma420LocType is equal to 0 to 5, inclusive, from left to right

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

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

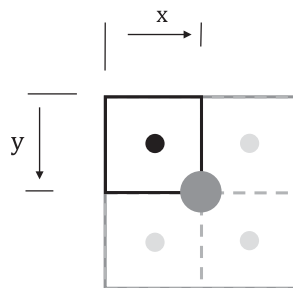


Figure 5 — Location of the top-left chroma sample when ChromaFormatIdc is equal to 1 (4:2:0 chroma format) when Chroma420LocType is equal to 1

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

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

When ChromaFormatIdc is equal to 1 (4:2:0 chroma format) and the decoded video content is intended for interpretation according to Rec. ITU-R BT.2020 or Rec. ITU-R BT.2100, vui_chroma_loc_info_present_flag should be equal to 1, and vui_chroma_sample_loc_type_frame, vui_chroma_sample_loc_type_top_field, and vui_chroma_sample_loc_type_bottom_field (as applicable) should be equal to 2.

8 SEI messages

8.1 General

[Clause 8](#) specifies the syntax and semantics for SEI messages.

For SEI messages for which the specified syntax structure is empty, such as the dependent random access point SEI message, the mere indication that the SEI message is present (e.g., as indicated by a payload type indicator) is sufficient to convey the associated information (e.g., by indicating that a set of specified constraints are fulfilled).

The semantics and persistence scope for each SEI message are specified in the semantics specification for each particular SEI message.

NOTE Persistence information for SEI messages is summarized in [Table 4](#).

Table 4 — Persistence scope of SEI messages

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

Table 4 (continued)

SEI message	Persistence scope
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

In the semantics of a particular SEI message, the phrase "the current layer" in the semantics refer to the layer that the particular SEI message is associated with, the phrase "the current picture" refer to the picture that the particular SEI message is associated with, and the phrase "the current CLVS" or "the CLVS" refers to the CLVS containing the current picture. The association of an SEI message to a layer or a picture is specified in a video coding specification that specifies a coded video bitstream with which the SEI messages are used.

The values of some SEI message syntax elements, including `fp_arrangement_id` and `omni_viewport_id`, are split into two sets of value ranges, where the first set is specified as "may be used as determined by the application", and the second set is specified as "reserved for future use by ITU-T | ISO/IEC". Applications should be cautious of potential "collisions" of the interpretation for values of these syntax elements belonging to the first set of value ranges. Since different applications might use these IDs having values in the first set of value ranges for different purposes, particular care should be exercised in the design of encoders that generate SEI messages with these IDs having values in the first set of value ranges, and in the design of decoders that interpret SEI messages with these IDs. This document does not define any management for these values. These IDs having values in the first set of value ranges might only be suitable for use in contexts in which "collisions" of usage (i.e., different definitions of the syntax and semantics of an SEI message with one of these IDs having the same value in the first set of value ranges) 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.

Some SEI messages include a persistency cancel flag in the syntax. For example, the film grain characteristics SEI message includes the `fg_characteristics_cancel_flag`. Regardless of whether such an SEI message is included in an NAL unit that precedes or succeeds the VCL NAL units of the current picture (i.e., the picture associated with the SEI message), the current picture associated with the SEI message is not included in the persistence established by any previous SEI message of that type in decoding order.

8.2 Filler payload SEI message

8.2.1 Filler payload SEI message syntax

<code>filler_payload(payloadSize) {</code>	Descriptor
<code> for(k = 0; k < payloadSize; k++)</code>	
<code> ff_byte /* equal to 0xFF */</code>	<code>f(8)</code>
<code>}</code>	

8.2.2 Filler payload SEI message semantics

This SEI message contains a series of `payloadSize` bytes of value 0xFF, which can be discarded.

ff_byte shall be a byte having the value 0xFF.

8.3 User data registered by Rec. ITU-T T.35 SEI message

8.3.1 User data registered by Rec. ITU-T T.35 SEI message syntax

user_data_registered_itu_t_t35(payloadSize) {	Descriptor
itu_t_t35_country_code	b(8)
if(itu_t_t35_country_code != 0xFF)	
i = 1	
else {	
itu_t_t35_country_code_extension_byte	b(8)
i = 2	
}	
do {	
itu_t_t35_payload_byte	b(8)
i++	
} while(i < payloadSize)	
}	

8.3.2 User data registered by Rec. ITU-T T.35 SEI message semantics

This SEI message contains user data registered as specified in Rec. ITU-T T.35, the contents of which are not specified in this document.

itu_t_t35_country_code shall be a byte having a value specified as a country code by Rec. ITU-T T.35:2000, Annex A.

itu_t_t35_country_code_extension_byte shall be a byte having a value specified as a country code by Rec. ITU-T T.35:2000, Annex B.

itu_t_t35_payload_byte shall be a byte containing data registered as specified in Rec. ITU-T T.35.

The Rec ITU-T T.35 terminal provider code and terminal provider oriented code shall be contained in the first one or more bytes of the **itu_t_t35_payload_byte**, in the format specified by the Administration that issued the terminal provider code. Any remaining **itu_t_t35_payload_byte** data shall be data having syntax and semantics as specified by the entity identified by the Rec ITU-T T.35 country code and terminal provider code.

8.4 User data unregistered SEI message

8.4.1 User data unregistered SEI message syntax

user_data_unregistered(payloadSize) {	Descriptor
uuid_iso_iec_11578	u(128)
for(i = 16; i < payloadSize; i++)	
user_data_payload_byte	b(8)
}	

8.4.2 User data unregistered SEI message semantics

This SEI message contains unregistered user data identified by a universal unique identifier (UUID), the contents of which are not specified in this document.

uuid_iso_iec_11578 shall have a value specified as a UUID according to the procedures of ISO/IEC 11578:1996, Annex A.

user_data_payload_byte shall be a byte containing data having syntax and semantics as specified by the UUID generator.

8.5 Film grain characteristics SEI message

8.5.1 Film grain characteristics SEI message syntax

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)
}	
}	

8.5.2 Film grain characteristics SEI message semantics

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

NOTE 1 For example, an encoder can 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.
- 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 film grain models specified in the film grain characteristics SEI message are expressed for application to decoded pictures that have 4:4:4 colour format 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 the decoded video is not 4:4:4 or the decoded video uses 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 the decoded pictures 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 edition of this document. Decoders shall ignore film grain characteristic SEI messages with **fg_model_id** equal to 2 or 3.

Table 5 — 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 equations in this subclause 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 equations in this subclause 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 BitDepth_Y (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 equations in this subclause is also greater than BitDepth_Y . In such a case, the input image $\hat{I}$ corresponding to an actual decoded picture would be generated by converting the actual decoded picture to be in the same colour representation domain as the 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 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 **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 **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 [subclause 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 [subclause 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 [subclause 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 [subclause 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_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 edition of this document. Decoders shall ignore film grain characteristic 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 $\text{fgBitDepth}[c]$ 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 $c = 0..2$, $x = 0..\text{PicWidthInLumaSamples} - 1$, and $y = 0..\text{PicHeightInLumaSamples} - 1$.

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 $c = 0..2$, $x = 0..PicWidthInLumaSamples - 1$, $y = 0..PicHeightInLumaSamples - 1$, 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 (*x*_B, *y*_B) of the current 8x8 block *b* that contains the sample value $\hat{I}[c][x][y]$ is derived as (*x*_B, *y*_B) = (*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 +=  $\hat{I}[ c ][ x_B * 8 + i ][ y_B * 8 + j ]$ 
bavg = 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( bavg >= fg_intensity_interval_lower_bound[ c ][ i ] &&
        bavg <= 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 ) ? ( Î[ c ][ x ][ y ] :
    Clip3( 0, 255, ( Î[ 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 equations in the remainder of this subclause, the variable s_j in each instance of the list fg_comp_model_value[c][s_j] is the value of intensityIntervalIdx[c][x][y][j] derived for the sample value Î[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 subclause:

- If fg_model_id is equal to 0, fg_comp_model_value[c][i][j] shall be in the range of 0 to 2^{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^(fgBitDepth[c] - 1) to 2^(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..PicWidthInLumaSamples - 1, and y = 0..PicHeightInLumaSamples - 1 as specified by:
- If fg_model_id is equal to 1, a frequency filtering model enables simulating the original film grain for c = 0..2, x = 0..PicWidthInLumaSamples - 1, and y = 0..PicHeightInLumaSamples - 1 as specified by:

$$\begin{aligned}
 G[c][x][y] = & (fg_comp_model_value[c][s_j][0] * Q[c][x][y] + \\
 & fg_comp_model_value[c][s_j][5] * \\
 & G[c - 1][x][y]) >> fg_log2_scale_factor
 \end{aligned}$$

(26)

where Q[c] is a two-dimensional random process generated by filtering 16x16 blocks gaussRv with random-valued elements gaussRv_{ij} 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 equation 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 uRv₀ and uRv₁, using the Box-Muller transformation specified by:

$$\text{gaussRv}_{ij} = \text{Sqrt}(-2 * \text{Ln}(\text{uRv}_0)) * \text{Cos}(2 * \pi * \text{uRv}_1)$$

(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 )

```

(28)

where IDCT16x16(z) refers to a unitary inverse discrete cosine transformation (IDCT) operating on a 16x16 matrix argument z as specified by:

$$\text{IDCT16x16}(z) = r * z * r^T \quad (29)$$

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

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

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 $c = 0..2$, $x = 0..\text{PicWidthInLumaSamples} - 1$, and $y = 0..\text{PicHeightInLumaSamples} - 1$ as specified by:

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

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 equation 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 Formulae 26 through 31.

$fg_comp_model_value[c][i][1]$ provides the second model value for the model as specified by fg_model_id . When fg_model_id is equal to 0, $fg_comp_model_value[c][i][1]$ 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.

8.6 Frame packing arrangement SEI message

8.6.1 Frame packing arrangement SEI message syntax

frame_packing_arrangement(payloadSize) {	Descriptor
fp_arrangement_id	ue(v)
fp_arrangement_cancel_flag	u(1)
if(!fp_arrangement_cancel_flag) {	
fp_arrangement_type	u(7)
fp_quincunx_sampling_flag	u(1)
fp_content_interpretation_type	u(6)
fp_spatial_flipping_flag	u(1)
fp_frame0_flipped_flag	u(1)
fp_field_views_flag	u(1)
fp_current_frame_is_frame0_flag	u(1)
fp_frame0_self_contained_flag	u(1)
fp_frame1_self_contained_flag	u(1)
if(!fp_quincunx_sampling_flag && fp_arrangement_type != 5) {	

fp_frame0_grid_position_x	u(4)
fp_frame0_grid_position_y	u(4)
fp_frame1_grid_position_x	u(4)
fp_frame1_grid_position_y	u(4)
}	
fp_arrangement_reserved_byte	u(8)
fp_arrangement_persistence_flag	u(1)
}	
fp_upsampled_aspect_ratio_flag	u(1)
}	

8.6.2 Frame packing arrangement SEI message semantics

This SEI message informs the decoder that the cropped decoded picture contains samples of multiple distinct spatially packed constituent frames that are packed into one frame, or that the output cropped decoded pictures in output order form a temporal interleaving of alternating first and second constituent frames, using an indicated frame packing arrangement scheme. This information can be used by the decoder to appropriately rearrange the samples and process the samples of the constituent frames appropriately for display or other purposes (which are outside the scope of this document).

NOTE 1 The interpretation of `frame_packing_arrangement_type` is in alignment with the code point specifications in Rec. ITU-T H.273 | ISO/IEC 23091-2. However, more values of `frame_packing_arrangement_type` are specified in Rec. ITU-T H.273 | ISO/IEC 23091-2 than are specified for use herein.

This SEI message may be associated with pictures that are either frames or fields (as determined outside the scope of this document). The frame packing arrangement of the samples is specified in terms of the sampling structure of a frame in order to define a frame packing arrangement structure that is invariant with respect to whether a picture is a single field of such a packed frame or is a complete packed frame.

fp_arrangement_id contains an identifying number that may be used to identify the usage of the frame packing arrangement SEI message. The value of `fp_arrangement_id` shall be in the range of 0 to $2^{32} - 2$, inclusive.

Values of `fp_arrangement_id` from 0 to 255, inclusive, and from 512 to $2^{31} - 1$, inclusive, may be used as determined by the application. Values of `fp_arrangement_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 encountering a value of `fp_arrangement_id` in the range of 256 to 511, inclusive, or in the range of 2^{31} to $2^{32} - 2$, inclusive, shall ignore it.

fp_arrangement_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous frame packing arrangement SEI message in output order that applies to the current layer. `fp_arrangement_cancel_flag` equal to 0 indicates that frame packing arrangement information follows.

fp_arrangement_type identifies the indicated interpretation of the sample arrays of the output cropped decoded picture as specified in [Table 7](#).

When `fp_arrangement_type` is equal to 3 or 4, each component plane of the output cropped decoded picture contains all samples (when `ffi_field_pic_flag` is equal to 0) or the samples corresponding to the top or bottom field (when `ffi_field_pic_flag` is equal to 1) of the samples of a frame packing arrangement structure.

Table 7 — Definition of `fp_arrangement_type`

Value	Interpretation
3	The frame packing arrangement structure contains a side-by-side packing arrangement of corresponding planes of two constituent frames as illustrated in Figure 6 , Figure 7 and Figure 10 .
4	The frame packing arrangement structure contains a top-bottom packing arrangement of corresponding planes of two constituent frames as illustrated in Figure 8 and Figure 9 .
5	The component planes of the output cropped decoded pictures in output order form a temporal interleaving of alternating first and second constituent frames as illustrated in Figure 11 .

NOTE 2 [Figure 6](#) to [Figure 10](#) provide typical examples of rearrangement and upconversion processing for various packing arrangement schemes. Actual characteristics of the constituent frames are signalled in detail by the subsequent syntax elements of the frame packing arrangement SEI message. In [Figure 6](#) to [Figure 10](#), an upconversion processing is performed on each constituent frame to produce frames having the same resolution as that of the decoded frame. An example of the upsampling method to be applied to a quincunx sampled frame as shown in [Figure 10](#) is to fill in missing positions with an average of the available spatially neighbouring samples (the average of the values of the available samples above, below, to the left and to the right of each sample to be generated). The actual upconversion process to be performed, if any, is outside the scope of this document.

NOTE 3 When the output time of the samples of constituent frame 0 differs from the output time of the samples of constituent frame 1 (i.e., when `fp_field_views_flag` is equal to 1 or `fp_arrangement_type` is equal to 5) and the display system in use presents two views simultaneously, the display time for constituent frame 0 could be delayed to coincide with the display time for constituent frame 1. (The display process is not specified in this document.)

NOTE 4 When `fp_field_views_flag` is equal to 1 or `fp_arrangement_type` is equal to 5, the value 0 for `fixed_pic_rate_within_cvs_flag` is not expected to be prevalent in industry use of this SEI message.

NOTE 5 `fp_arrangement_type` equal to 5 describes a temporal interleaving process of different views.

All other values of `fp_arrangement_type` are reserved for future use by ITU-T | ISO/IEC. It is a requirement of bitstream conformance that bitstreams conforming to this edition of this document shall not contain such other values of `fp_arrangement_type`. Decoders shall ignore frame packing arrangement SEI messages that contain reserved values of `fp_arrangement_type`.

`fp_quincunx_sampling_flag` equal to 1 indicates that each colour component plane of each constituent frame is quincunx sampled as illustrated in [Figure 10](#) and `fp_quincunx_sampling_flag` equal to 0 indicates that the colour component planes of each constituent frame are not quincunx sampled.

When `fp_arrangement_type` is equal to 5, it is a requirement of bitstream conformance that `fp_quincunx_sampling_flag` shall be equal to 0.

NOTE 6 For any chroma format (monochrome, 4:2:0, 4:2:2 or 4:4:4), the luma plane and each chroma plane (as applicable) is quincunx sampled as illustrated in [Figure 10](#) when `fp_quincunx_sampling_flag` is equal to 1.

`fp_content_interpretation_type` indicates the intended interpretation of the constituent frames as specified in [Table 8](#). Values of `fp_content_interpretation_type` that do not appear in [Table 8](#) are reserved for future specification by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this edition of this document. Decoders shall ignore frame packing arrangement SEI messages that contain reserved values of `fp_content_interpretation_type`.

For each specified frame packing arrangement scheme, there are two constituent frames that are referred to as frame 0 and frame 1.

Table 8 — Definition of `fp_content_interpretation_type`

Value	Interpretation
0	Unknown or unspecified relationship between the frame packed constituent frames
1	Indicates that the two constituent frames form the left and right views of a stereo view scene, with frame 0 being associated with the left view and frame 1 being associated with the right view
2	Indicates that the two constituent frames form the right and left views of a stereo view scene, with frame 0 being associated with the right view and frame 1 being associated with the left view