
**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*

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23002-7:2022



STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23002-7:2022



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

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.....	1
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.....	12
5.10 Variables, syntax elements and tables.....	13
5.11 Text description of logical operations.....	14
5.12 Processes.....	15
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.....	28
8.3.1 User data registered by Rec. ITU-T T.35 SEI message syntax.....	28
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.....	29
8.5.1 Film grain characteristics SEI message syntax.....	29
8.5.2 Film grain characteristics SEI message semantics.....	30
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.....	38
8.7 Parameter sets inclusion indication SEI message.....	47
8.7.1 Parameter sets inclusion indication SEI message syntax.....	47
8.7.2 Parameter sets inclusion indication SEI message semantics.....	47
8.8 Decoded picture hash SEI message.....	48
8.8.1 Decoded picture hash SEI message syntax.....	48
8.8.2 Decoded picture hash SEI message semantics.....	48
8.9 Mastering display colour volume SEI message.....	50

8.9.1	Mastering display colour volume SEI message syntax	50
8.9.2	Mastering display colour volume SEI message semantics	50
8.10	Content light level information SEI message	52
8.10.1	Content light level information SEI message syntax	52
8.10.2	Content light level information SEI message semantics	52
8.11	Dependent random access point indication SEI message	53
8.11.1	Dependent random access point indication SEI message syntax	53
8.11.2	Dependent random access point indication SEI message semantics	53
8.12	Alternative transfer characteristics information SEI message	54
8.12.1	Alternative transfer characteristics information SEI message syntax	54
8.12.2	Alternative transfer characteristics SEI message semantics	54
8.13	Ambient viewing environment SEI message	54
8.13.1	Ambient viewing environment SEI message syntax	54
8.13.2	Ambient viewing environment SEI message semantics	54
8.14	Content colour volume SEI message	55
8.14.1	Content colour volume SEI message syntax	55
8.14.2	Content colour volume SEI message semantics	56
8.15	Omnidirectional video specific SEI messages	58
8.15.1	Sample location remapping process	58
8.15.2	Equirectangular projection SEI message	69
8.15.3	Generalized cubemap projection SEI message	72
8.15.4	Sphere rotation SEI message	78
8.15.5	Region-wise packing SEI message	79
8.15.6	Omnidirectional viewport SEI message	86
8.16	Frame-field information SEI message	88
8.16.1	Frame-field information SEI message syntax	88
8.16.2	Frame-field information SEI message semantics	88
8.17	Sample aspect ratio information SEI message	91
8.17.1	Sample aspect ratio information SEI message syntax	91
8.17.2	Sample aspect ratio information SEI message semantics	91
8.18	Annotated regions SEI message	92
8.18.1	Annotated regions SEI message syntax	92
8.18.2	Annotated regions SEI message semantics	94
8.19	Scalability dimension information SEI message	97
8.19.1	Scalability dimension information SEI message syntax	97
8.19.2	Scalability dimension information SEI message semantics	97
8.20	Multiview acquisition information SEI message	99
8.20.1	Multiview acquisition information SEI message syntax	99
8.20.2	Multiview acquisition information SEI message semantics	100
8.21	Multiview view position SEI message	104
8.21.1	Multiview view position SEI message syntax	104
8.21.2	Multiview view position SEI message semantics	104
8.22	Depth representation information SEI message	104
8.22.1	Depth representation information SEI message syntax	104
8.22.2	Depth representation information SEI message semantics	105
8.23	Alpha channel information SEI message	108
8.23.1	Alpha channel information SEI message syntax	108
8.23.2	Alpha channel information SEI message semantics	109
8.24	Extended DRAP indication SEI message	112
8.24.1	Extended DRAP indication SEI message syntax	112
8.24.2	Extended DRAP indication SEI message semantics	112
8.25	Display orientation SEI message	113
8.25.1	Display orientation SEI message syntax	113
8.25.2	Display orientation SEI message semantics	113
8.26	Colour transform information SEI message	114
8.26.1	Colour transform information SEI message syntax	114
8.26.2	Colour transform information SEI message semantics	115
8.27	Reserved SEI message	119

8.27.1	Reserved SEI message syntax.....	119
8.27.2	Reserved SEI message semantics.....	119
9	Parsing process for k-th order Exp-Golomb codes.....	119
9.1	General.....	119
9.2	Mapping process for signed Exp-Golomb codes.....	121
Bibliography.....		122

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23002-7:2022

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).

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

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. 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 information*, in collaboration with ITU-T (as Rec. ITU-T H.274).

This second edition cancels and replaces the first edition (ISO/IEC 23002-7:2021), which has been technically revised.

The main changes are as follows:

- the addition of the annotated regions SEI message,
- the addition of the alpha channel information SEI message,
- the addition of the depth representation information SEI message,
- the addition of the multiview acquisition information SEI message,
- the addition of the multiview view position SEI message,
- the addition of the scalability dimension information SEI message,
- the addition of the extended dependent random access point indication SEI message,
- the addition of the display orientation SEI message, and
- the addition of the colour transform information 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 (the current version) 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 ISO/IEC as ISO/IEC 23002-7:202X corresponds to the second version. At the time of publication of this edition by ISO/IEC, a corresponding second edition of Rec. ITU-T H.274 was in preparation for publication by ITU-T.

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.

Patent declarations

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents or <https://patents.iec.ch>.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23002-7:2022

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)*

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

Recommendation 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 5646, *Tags for Identifying Languages.*

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 *PIs* 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 *PIs* of the same layer that consists, in *decoding order*, of a *CLVSS PU*, followed by zero or more *PIs* that are not *CLVSS PUs*, including all subsequent *PIs* 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 CLVSS 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
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
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.
÷	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.
- ^ 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

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

$\text{Asin}(x)$ 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)

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

Ceil(x) smallest integer greater than or equal to x (5)

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

Cos(x) trigonometric cosine function operating on an argument x in units of radians (7)

Floor(x) largest integer less than or equal to x (8)

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

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

Round(x) = Sign(x) * Floor(Abs(x) + 0.5) (11)

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x = 0 \\ -1 & ; \quad x < 0 \end{cases} \quad (12)$$

Sin(x) trigonometric sine function operating on an argument x in units of radians (13)

Sqrt(x) square root of x (14)

Tan(x) trigonometric tangent function operating on an argument x in units of radians (15)

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.

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 (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

	Descriptor
<code>vui_parameters(payloadSize) {</code>	
vui_progressive_source_flag	u(1)
vui_interlaced_source_flag	u(1)
vui_non_packed_constraint_flag	u(1)
vui_non_projected_constraint_flag	u(1)
vui_aspect_ratio_info_present_flag	u(1)
if(<code>vui_aspect_ratio_info_present_flag</code>) {	
vui_aspect_ratio_constant_flag	u(1)
vui_aspect_ratio_idc	u(8)
if(<code>vui_aspect_ratio_idc == 255</code>) {	
vui_sar_width	u(16)
vui_sar_height	u(16)
}	
}	
vui_overscan_info_present_flag	u(1)
if(<code>vui_overscan_info_present_flag</code>)	
vui_overscan_appropriate_flag	u(1)
vui_colour_description_present_flag	u(1)
if(<code>vui_colour_description_present_flag</code>) {	
vui_colour_primaries	u(8)
vui_transfer_characteristics	u(8)
vui_matrix_coefs	u(8)
vui_full_range_flag	u(1)
}	
vui_chroma_loc_info_present_flag	u(1)
if(<code>vui_chroma_loc_info_present_flag</code>) {	
if(<code>vui_progressive_source_flag && !vui_interlaced_source_flag</code>)	
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 `BitDepthY`, and when `ChromaFormatIdc` is not equal to 0, a bit depth for the samples of the two associated chroma components, denoted herein by `BitDepthC`.

Table 2 — SubWidthC and SubHeightC values derived from ChromaFormatIdc

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

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

- If `vui_progressive_source_flag` is equal to 1 and `vui_interlaced_source_flag` is equal to 0, the source scan type of the pictures should be interpreted as progressive only.
- Otherwise, if `vui_progressive_source_flag` is equal to 0 and `vui_interlaced_source_flag` is equal to 1, the source scan type of the pictures should be interpreted as interlaced only.
- Otherwise, if `vui_progressive_source_flag` is equal to 0 and `vui_interlaced_source_flag` is equal to 0, the source scan type of the pictures should be interpreted as unknown or unspecified or specified by external means not specified in this 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.

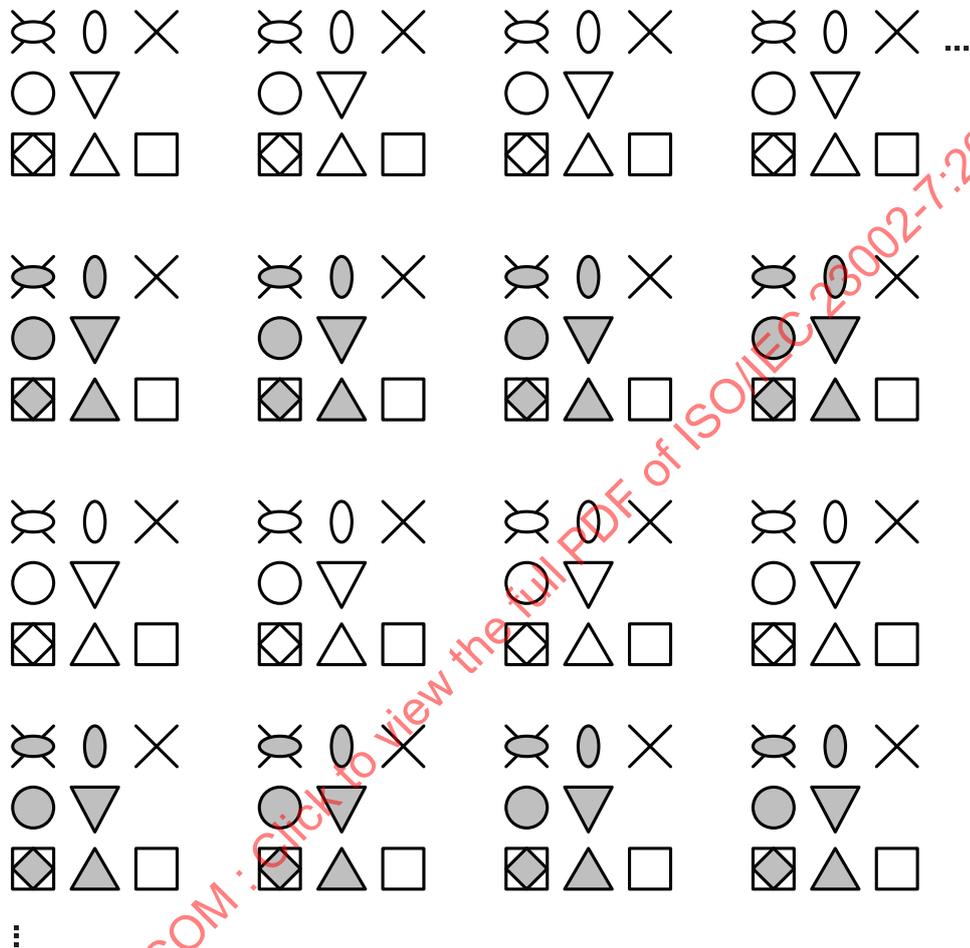
vui_chroma_loc_info_present_flag equal to 1 specifies that either `vui_chroma_sample_loc_type_frame` or both `vui_chroma_sample_loc_type_top_field` and `vui_chroma_sample_loc_type_bottom_field` are present. `vui_chroma_loc_info_present_flag` equal to 0 specifies that `vui_chroma_sample_loc_type_frame`, `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](#).

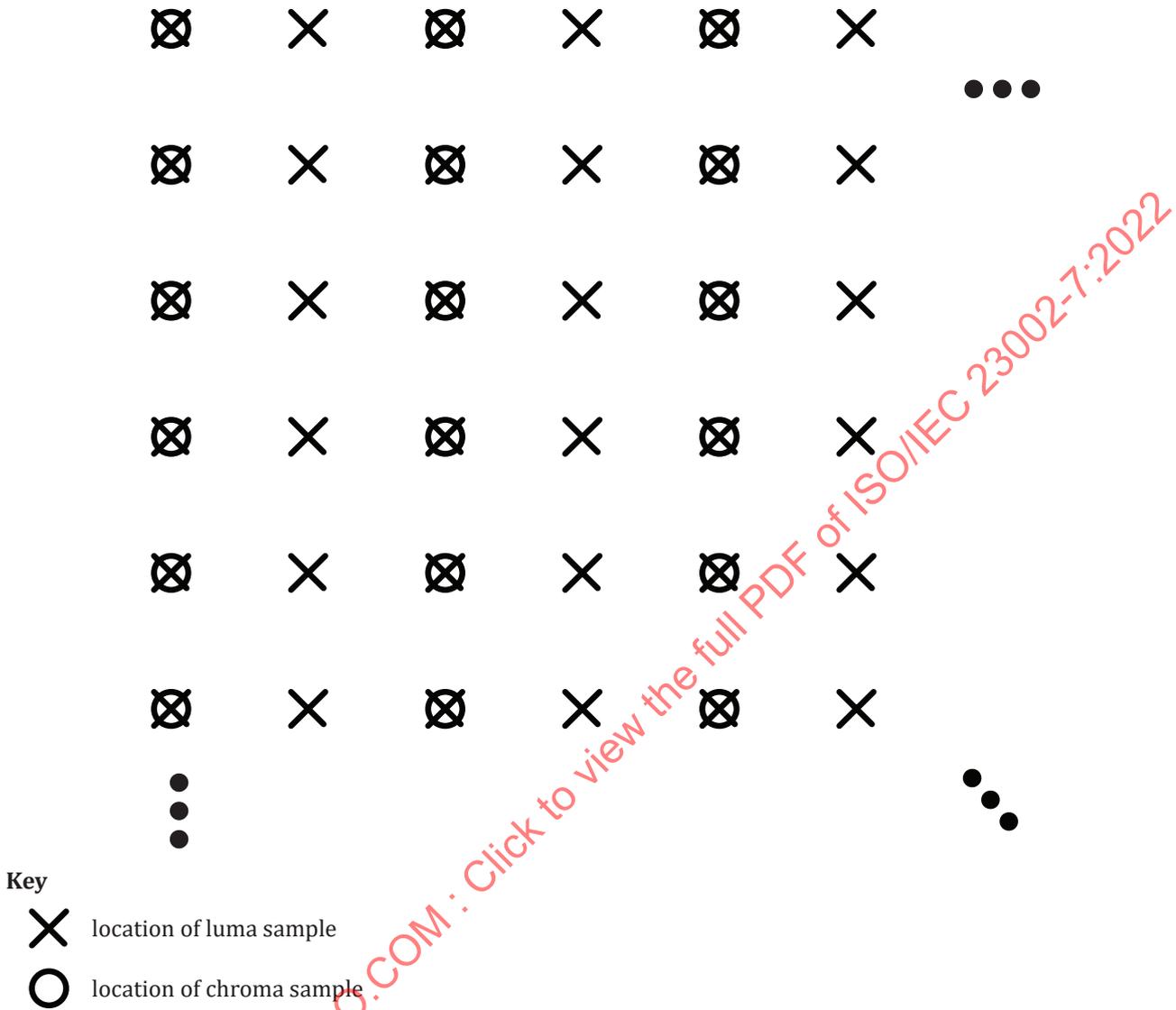


Figure 2 — Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a picture

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](#).

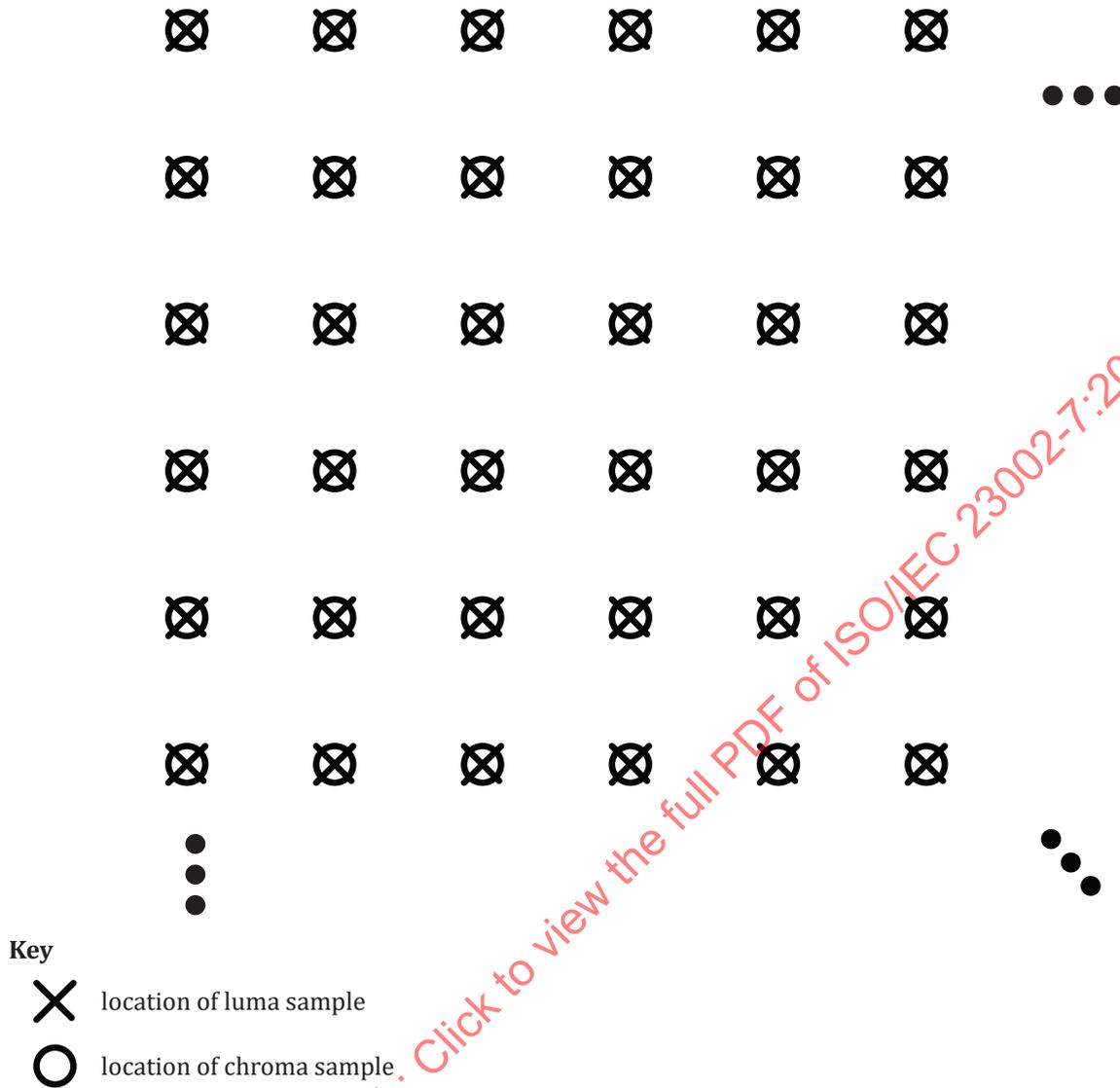


Figure 3 — Nominal vertical and horizontal locations of 4:4:4 luma and chroma samples in a picture

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.

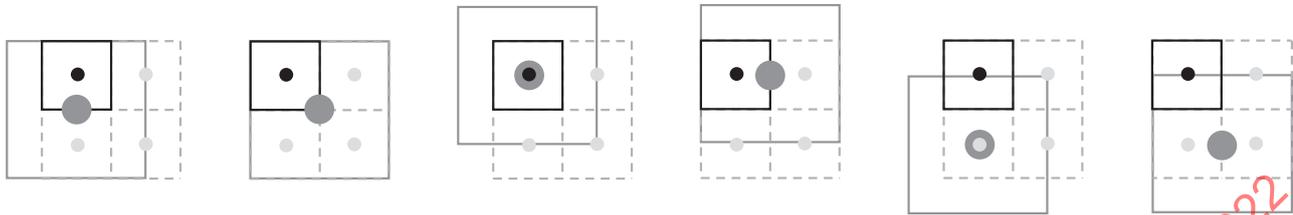


Figure 4 — Location of the top-left chroma sample when ChromaFormatIdc is equal to 1 (4:2:0 chroma format) and Chroma420LocType is equal to 0 to 5, inclusive, from left to right

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.

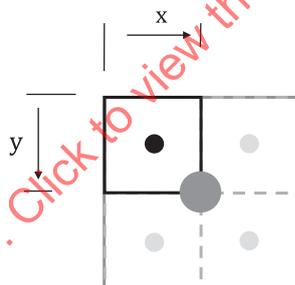


Figure 5 — Location of the top-left chroma sample when ChromaFormatIdc is equal to 1 (4:2:0 chroma format) when Chroma420LocType is equal to 1

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
Display orientation	Specified by the syntax of the SEI message
Referenced parameter sets	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
Colour transform information	Specified by the syntax of 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
Alpha channel information	Specified by the syntax of the SEI message
Frame-field information	The PU containing the SEI message
Depth representation information	Specified by the semantics of the SEI message
Multiview acquisition information	The CVS containing the SEI message
Multiview view position	The CVS containing the SEI message
Annotated regions	Specified by the syntax of the SEI message
Sample aspect ratio information	Specified by the syntax of the SEI message

Table 4 (continued)

SEI message	Persistence scope
Scalability dimension information	The CVS containing the SEI message
Extended DRAP indication	The picture associated with 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.

8.2 Filler payload SEI message

8.2.1 Filler payload SEI message syntax

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

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(<code>itu_t_t35_country_code</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 film grain synthesis.

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 can be the decoded pictures or converted from the decoded pictures, for the display process is optional and does not need to 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 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:

- Apicturewidthandpictureheightinunitsoflumasamples,denotedhereinbyPicWidthInLumaSamples 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 $\text{fgBitDepth}[0]$ is derived as follows:

$$\text{fgBitDepth}[0] = \text{fg_bit_depth_luma_minus8} + 8 \quad (16)$$

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

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

$$\text{fgBitDepth}[c] = \text{fg_bit_depth_chroma_minus8} + 8, \text{ with } c = 1, 2 \quad (17)$$

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:

$$I_{\text{grain}}[c][x][y] = \text{Clip3}(0, (1 \ll \text{fgBitDepth}[c]) - 1, \hat{I}[c][x][y] + G[c][x][y]) \quad (18)$$

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

$$I_{\text{grain}}[c][x][y] = \text{Clip3}(0, (1 \ll \text{fgBitDepth}[c]) - 1, \hat{I}[c][x][y] + \text{Round}(\hat{I}[c][x][y] * G[c][x][y]) \div ((1 \ll \text{fgBitDepth}[c]) - 1)) \quad (19)$$

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

$y = 0..PicHeightInLumaSamples - 1$, 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 (xB, yB) of the current 8x8 block b that contains the sample value $\hat{I}[c][x][y]$ is derived as $(xB, yB) = (x / 8, y / 8)$.
 - The average value b_{avg} of the current 8x8 block b is derived as follows:

$$\begin{aligned}
 & sum8x8 = 0 \\
 & for(i = 0; i < 8; i++) \\
 & \quad for(j = 0; j < 8; j++) \\
 & \quad \quad sum8x8 += \hat{I}[c][xB * 8 + i][yB * 8 + j] \\
 & b_{avg} = Clip3(0, 255, (sum8x8 + (1 \ll (fgBitDepth[c] - 3))) \gg (fgBitDepth[c] - 2))
 \end{aligned} \tag{20}$$

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

$$\begin{aligned}
 & for(i = 0, j = 0; i \leq fg_num_intensity_intervals_minus1[c]; i++) \\
 & \quad if(b_{avg} \geq fg_intensity_interval_lower_bound[c][i] \&\& \\
 & \quad \quad b_{avg} \leq fg_intensity_interval_upper_bound[c][i]) { \\
 & \quad \quad \quad intensityIntervalIdx[c][x][y][j] = i \\
 & \quad \quad \quad j++ \\
 & \quad \quad } \\
 & numApplicableIntensityIntervals[c][x][y] = j
 \end{aligned} \tag{21}$$

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

$$\begin{aligned}
 & I_g[c][x][y] = (fgBitDepth[c] == 8) ? (\hat{I}[c][x][y] : \\
 & \quad Clip3(0, 255, (\hat{I}[c][x][y] + (1 \ll (fgBitDepth[c] - 9))) \gg (fgBitDepth[c] - 8))) \\
 & for(i = 0, j = 0; i \leq fg_num_intensity_intervals_minus1[c]; i++) \\
 & \quad if(I_g[c][x][y] \geq fg_intensity_interval_lower_bound[c][i] \&\& \\
 & \quad \quad I_g[c][x][y] \leq fg_intensity_interval_upper_bound[c][i]) { \\
 & \quad \quad \quad intensityIntervalIdx[c][x][y][j] = i \\
 & \quad \quad \quad j++ \\
 & \quad \quad } \\
 & numApplicableIntensityIntervals[c][x][y] = j
 \end{aligned} \tag{22}$$

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 $\hat{I}[c][x][y]$.

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

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

$$G[c][x][y] = (fg_comp_model_value[c][s_j][0] * Q[c][x][y] + fg_comp_model_value[c][s_j][5] * G[c - 1][x][y]) \gg fg_log2_scale_factor \quad (23)$$

where $Q[c]$ is a two-dimensional random process generated by filtering 16×16 blocks $gaussRv$ with random-value 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 uRv_0 and uRv_1 , using the Box-Muller transformation specified by:

$$gaussRv_{i,j} = \text{Sqrt}(-2 * \text{Ln}(uRv_0)) * \text{Cos}(2 * \pi * uRv_1) \quad (24)$$

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 ][ s_j ][ 3 ] && y <
fg_comp_model_value[ c ][ s_j ][ 4 ] ) ||
      x > fg_comp_model_value[ c ][ s_j ][ 1 ] || y >
fg_comp_model_value[ c ][ s_j ][ 2 ] )
      gaussRv[ x ][ y ] = 0
filteredRv = IDCT16x16( gaussRv )

```

(25)

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

$$IDCT16x16(z) = r * z * r^T \quad (26)$$

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

$$r_{i,j} = \frac{ ((i = 0) ? 1 : \text{Sqrt}(2)) * \text{Cos}\left(\frac{ i * (2 * j + 1) * \pi }{ 32 } \right) }{ 4 } \quad (27)$$

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 $fg_comp_model_value[c][s_j][i]$ 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 $c = 0..2$, $x = 0..PicWidthInLumaSamples - 1$, and $y = 0..PicHeightInLumaSamples - 1$ as specified by:

$$\begin{aligned}
 G[c][x][y] = & (fg_comp_model_value[c][s_j][0] * n[c][x][y] + \\
 & fg_comp_model_value[c][s_j][1] * (G[c][x-1][y] + \\
 & ((fg_comp_model_value[c][s_j][4] * G[c][x][y-1]) \gg \\
 & fg_log2_scale_factor)) + fg_comp_model_value[c][s_j][3] * \\
 & (((fg_comp_model_value[c][s_j][4] * G[c][x-1][y-1]) \gg \\
 & fg_log2_scale_factor) + G[c][x+1][y-1]) + \\
 & fg_comp_model_value[c][s_j][5] * (G[c][x-2][y] + \\
 & ((fg_comp_model_value[c][s_j][4] * fg_comp_model_value[c][s_j][4] * \\
 & G[c][x][y-2]) \gg (2 * fg_log2_scale_factor))) + \\
 & fg_comp_model_value[c][s_j][2] * G[c-1][x][y]) \gg fg_log2_scale_factor
 \end{aligned} \tag{28}$$

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 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.

$fg_comp_model_value[c][i][0]$ provides the first model value for the model as specified by fg_model_id . $fg_comp_model_value[c][i][0]$ corresponds to the standard deviation of the Gaussian noise term in the generation functions specified in Formulae 23 through 28.

$fg_comp_model_value[c][i][1]$ 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

NOTE 7 The value 2 for `fp_content_interpretation_type` is not expected to be prevalent in industry use of this SEI message. However, the value was specified herein for purposes of completeness.

fp_spatial_flipping_flag equal to 1, when `fp_arrangement_type` is equal to 3 or 4, indicates that one of the two constituent frames is spatially flipped relative to its intended orientation for display or other such purposes.

When `fp_arrangement_type` is equal to 3 or 4 and `fp_spatial_flipping_flag` is equal to 1, the type of spatial flipping that is indicated is as follows:

- If `fp_arrangement_type` is equal to 3, the indicated spatial flipping is horizontal flipping.
- Otherwise (`fp_arrangement_type` is equal to 4), the indicated spatial flipping is vertical flipping.

When `fp_arrangement_type` is not equal to 3 or 4, it is a requirement of bitstream conformance that `fp_spatial_flipping_flag` shall be equal to 0. When `fp_arrangement_type` is not equal to 3 or 4, the value 1 for `fp_spatial_flipping_flag` is reserved for future use by ITU-T | ISO/IEC. When `fp_arrangement_type` is not equal to 3 or 4, decoders shall ignore the value 1 for `fp_spatial_flipping_flag`.

fp_frame0_flipped_flag, when `fp_spatial_flipping_flag` is equal to 1, indicates which one of the two constituent frames is flipped.

When `fp_spatial_flipping_flag` is equal to 1, `fp_frame0_flipped_flag` equal to 0 indicates that frame 0 is not spatially flipped and frame 1 is spatially flipped and `fp_frame0_flipped_flag` equal to 1 indicates that frame 0 is spatially flipped and frame 1 is not spatially flipped.

When `fp_spatial_flipping_flag` is equal to 0, it is a requirement of bitstream conformance that `fp_frame0_flipped_flag` shall be equal to 0. When `fp_spatial_flipping_flag` is equal to 0, the value 1 for `fp_spatial_flipping_flag` is reserved for future use by ITU-T | ISO/IEC. When `fp_spatial_flipping_flag` is equal to 0, decoders shall ignore the value of `fp_frame0_flipped_flag`.

`fp_field_views_flag` equal to 1 indicates that all pictures in the current CLVS are coded as fields, all fields of a particular parity are considered a first constituent frame and all fields of the opposite parity are considered a second constituent frame. It is a requirement of bitstream conformance that the `fp_field_views_flag` shall be equal to 0, the value 1 for `fp_field_views_flag` is reserved for future use by ITU-T | ISO/IEC and decoders shall ignore the value of `fp_field_views_flag`.

`fp_current_frame_is_frame0_flag` equal to 1, when `fp_arrangement` is equal to 5, indicates that the current decoded frame is constituent frame 0 and the next decoded frame in output order is constituent frame 1 and the display time of the constituent frame 0 should be delayed to coincide with the display time of constituent frame 1. `fp_current_frame_is_frame0_flag` equal to 0 when `fp_arrangement` is equal to 5, indicates that the current decoded frame is constituent frame 1 and the previous decoded frame in output order is constituent frame 0 and the display time of the constituent frame 1 should not be delayed for purposes of stereo-view pairing.

When `fp_arrangement_type` is not equal to 5, the constituent frame associated with the upper-left sample of the decoded frame is considered to be constituent frame 0 and the other constituent frame is considered to be constituent frame 1. When `fp_arrangement_type` is not equal to 5, it is a requirement of bitstream conformance that `fp_current_frame_is_frame0_flag` shall be equal to 0. When `fp_arrangement_type` is not equal to 5, the value 1 for `fp_current_frame_is_frame0_flag` is reserved for future use by ITU-T | ISO/IEC. When `fp_arrangement_type` is not equal to 5, decoders shall ignore the value of `fp_current_frame_is_frame0_flag`.

`fp_frame0_self_contained_flag` equal to 1 indicates that no inter prediction operations within the decoding process for the samples of constituent frame 0 of the CLVS refer to samples of any constituent frame 1. `fp_frame0_self_contained_flag` equal to 0 indicates that some inter prediction operations within the decoding process for the samples of constituent frame 0 of the CLVS might or might not refer to samples of some constituent frame 1. Within a CLVS, the value of `fp_frame0_self_contained_flag` in all frame packing arrangement SEI messages shall be the same.

`fp_frame1_self_contained_flag` equal to 1 indicates that no inter prediction operations within the decoding process for the samples of constituent frame 1 of the CLVS refer to samples of any constituent frame 0. `fp_frame1_self_contained_flag` equal to 0 indicates that some inter prediction operations within the decoding process for the samples of constituent frame 1 of the CLVS might or might not refer to samples of some constituent frame 0. Within a CLVS, the value of `fp_frame1_self_contained_flag` in all frame packing arrangement SEI messages shall be the same.

When `fp_quincunx_sampling_flag` is equal to 0 and `fp_arrangement_type` is not equal to 5, two (*x*, *y*) coordinate pairs are specified to determine the indicated luma sampling grid alignment for constituent frame 0 and constituent frame 1, relative to the upper left corner of the rectangular area represented by the samples of the corresponding constituent frame.

NOTE 8 The location of chroma samples relative to luma samples could be indicated by the `vui_chroma_sample_loc_type_frame` or `vui_chroma_sample_loc_type_top_field` and `vui_chroma_sample_loc_type_bottom_field` syntax elements in the VUI parameters, when present.

`fp_frame0_grid_position_x` (when present) specifies the *x* component of the (*x*, *y*) coordinate pair for constituent frame 0.

`fp_frame0_grid_position_y` (when present) specifies the *y* component of the (*x*, *y*) coordinate pair for constituent frame 0.

fp_frame1_grid_position_x (when present) specifies the x component of the (x, y) coordinate pair for constituent frame 1.

fp_frame1_grid_position_y (when present) specifies the y component of the (x, y) coordinate pair for constituent frame 1.

When `fp_quincunx_sampling_flag` is equal to 0 and `fp_arrangement_type` is not equal to 5 the (x, y) coordinate pair for each constituent frame is interpreted as follows:

- If the (x, y) coordinate pair for a constituent frame is equal to (0, 0), this indicates a default sampling grid alignment specified as follows:
 - If `fp_arrangement_type` is equal to 3, the indicated position is the same as for the (x, y) coordinate pair value (4, 8), as illustrated in [Figure 6](#).
 - Otherwise (`fp_arrangement_type` is equal to 4), the indicated position is the same as for the (x, y) coordinate pair value (8, 4), as illustrated in [Figure 8](#).
- Otherwise, if the (x, y) coordinate pair for a constituent frame is equal to (15, 15), this indicates that the sampling grid alignment is unknown or unspecified or specified by other means not specified in this document.
- Otherwise, the x and y elements of the (x, y) coordinate pair specify the indicated horizontal and vertical sampling grid alignment positioning to the right of and below the upper left corner of the rectangular area represented by the corresponding constituent frame, respectively, in units of one sixteenth of the luma sample grid spacing between the samples of the columns and rows of the constituent frame that are present in the decoded frame (prior to any upsampling for display or other purposes).

NOTE 9 The spatial location reference information `fp_frame0_grid_position_x`, `fp_frame0_grid_position_y`, `fp_frame1_grid_position_x`, and `fp_frame1_grid_position_y` is not provided when `fp_quincunx_sampling_flag` is equal to 1 because the spatial alignment in this case is assumed to be such that constituent frame 0 and constituent frame 1 cover corresponding spatial areas with interleaved quincunx sampling patterns as illustrated in [Figure 10](#).

fp_arrangement_reserved_byte is reserved for future use by ITU-T | ISO/IEC. It is a requirement of bitstream conformance that the value of `fp_arrangement_reserved_byte` shall be equal to 0. All other values of `fp_arrangement_reserved_byte` are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of `fp_arrangement_reserved_byte`.

fp_arrangement_persistence_flag specifies the persistence of the frame packing arrangement SEI message for the current layer.

`fp_arrangement_persistence_flag` equal to 0 specifies that the frame packing arrangement SEI message applies to the current decoded frame only.

`fp_arrangement_persistence_flag` equal to 1 specifies that the frame packing arrangement SEI message applies to the current decoded picture and persists 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 frame packing arrangement SEI message is output that follows the current picture in output order.

fp_upsampled_aspect_ratio_flag equal to 1 indicates that the SAR indicated by the VUI parameters or the SARI SEI message identifies the SAR of the samples after the application of an upconversion process to produce a higher resolution frame from each constituent frame as illustrated in [Figure 6](#) to [Figure 10](#). `fp_upsampled_aspect_ratio_flag` equal to 0 indicates that the SAR indicated by the VUI parameters or

the SARI SEI message identifies the SAR of the samples before the application of any such upconversion process.

NOTE 10 The SAR indicated in the VUI parameters or the SARI SEI message could indicate the preferred display picture shape for the packed decoded frame output by a decoder that does not interpret the frame packing arrangement SEI message. When `fp_upsampled_aspect_ratio_flag` is equal to 1, the SAR produced in each upconverted colour plane is indicated to be the same as the SAR indicated in the VUI parameters or the SARI SEI message in the examples shown in Figure 6 to Figure 10. When `fp_upsampled_aspect_ratio_flag` is equal to 0, the SAR produced in each colour plane prior to upconversion is indicated to be the same as the SAR indicated in the VUI parameters or the SARI SEI message in the examples shown in Figure 6 to Figure 10.

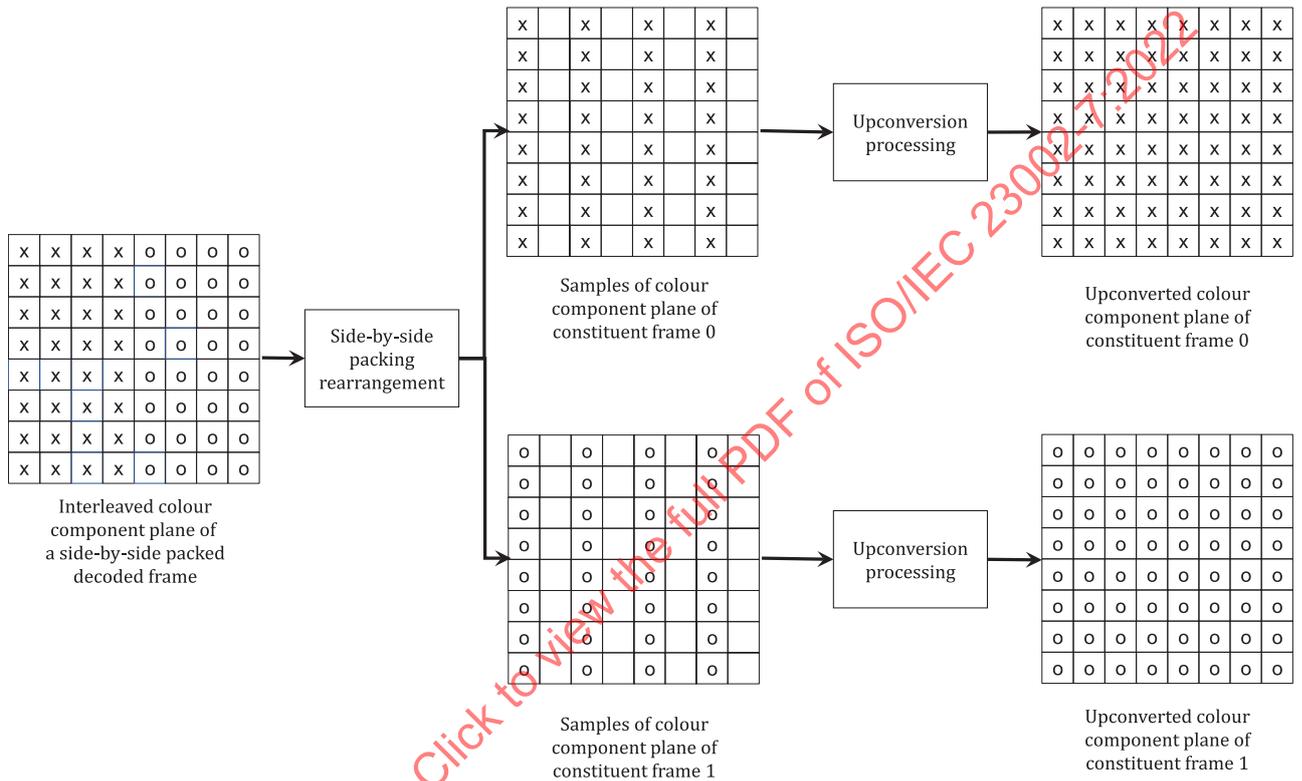


Figure 6 — Flowchart for rearrangement and upconversion of side-by-side packing arrangement with `fp_arrangement_type` equal to 3, `fp_quincunx_sampling_flag` equal to 0 and `(x,y)` equal to (0,0) or (4,8) for both constituent frames

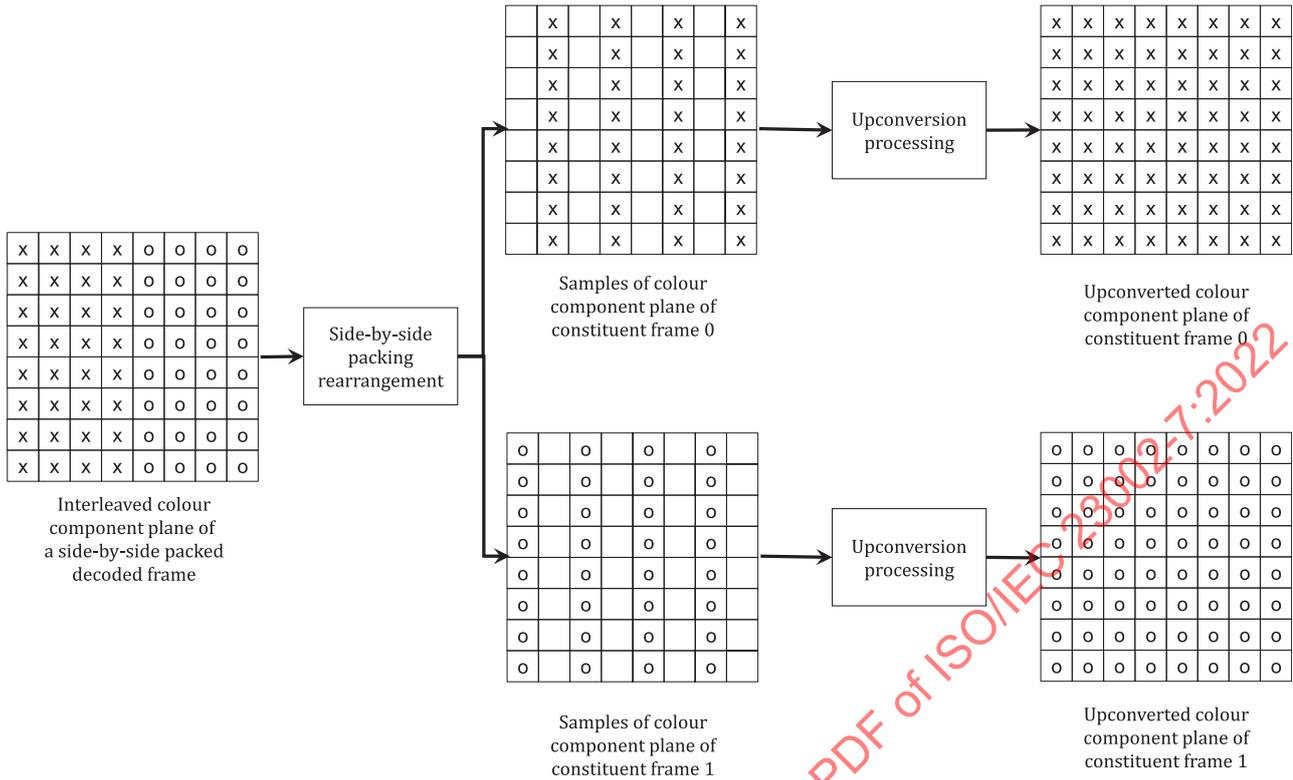


Figure 7 — Flowchart for rearrangement and upconversion of side-by-side packing arrangement with fp_arrangement_type equal to 3, fp_quincunx_sampling_flag equal to 0, (x, y) equal to (12, 8) for constituent frame 0 and (x, y) equal to (0, 0) or (4, 8) for constituent frame 1

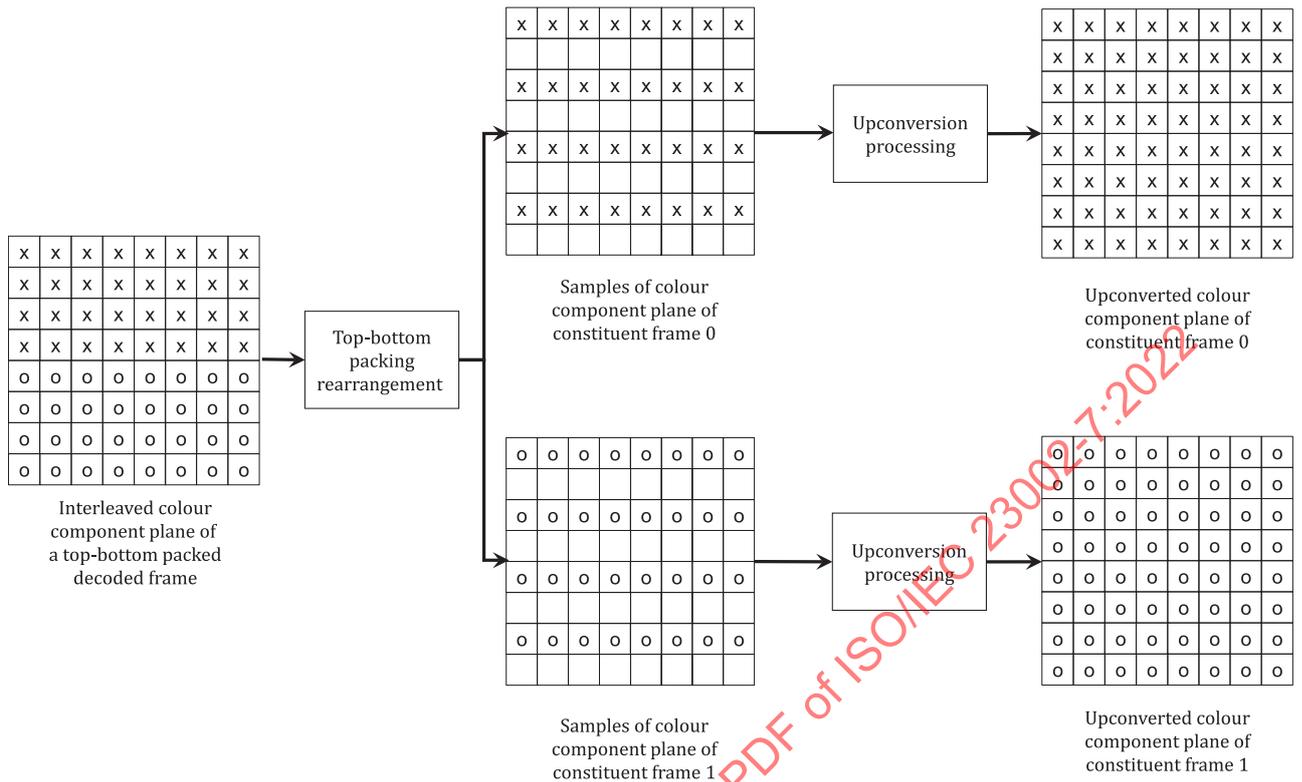


Figure 8 — Flowchart for rearrangement and upconversion of top-bottom packing arrangement with $fp_arrangement_type$ equal to 4, $fp_quincunx_sampling_flag$ equal to 0 and (x, y) equal to $(0, 0)$ or $(8, 4)$ for both constituent frames

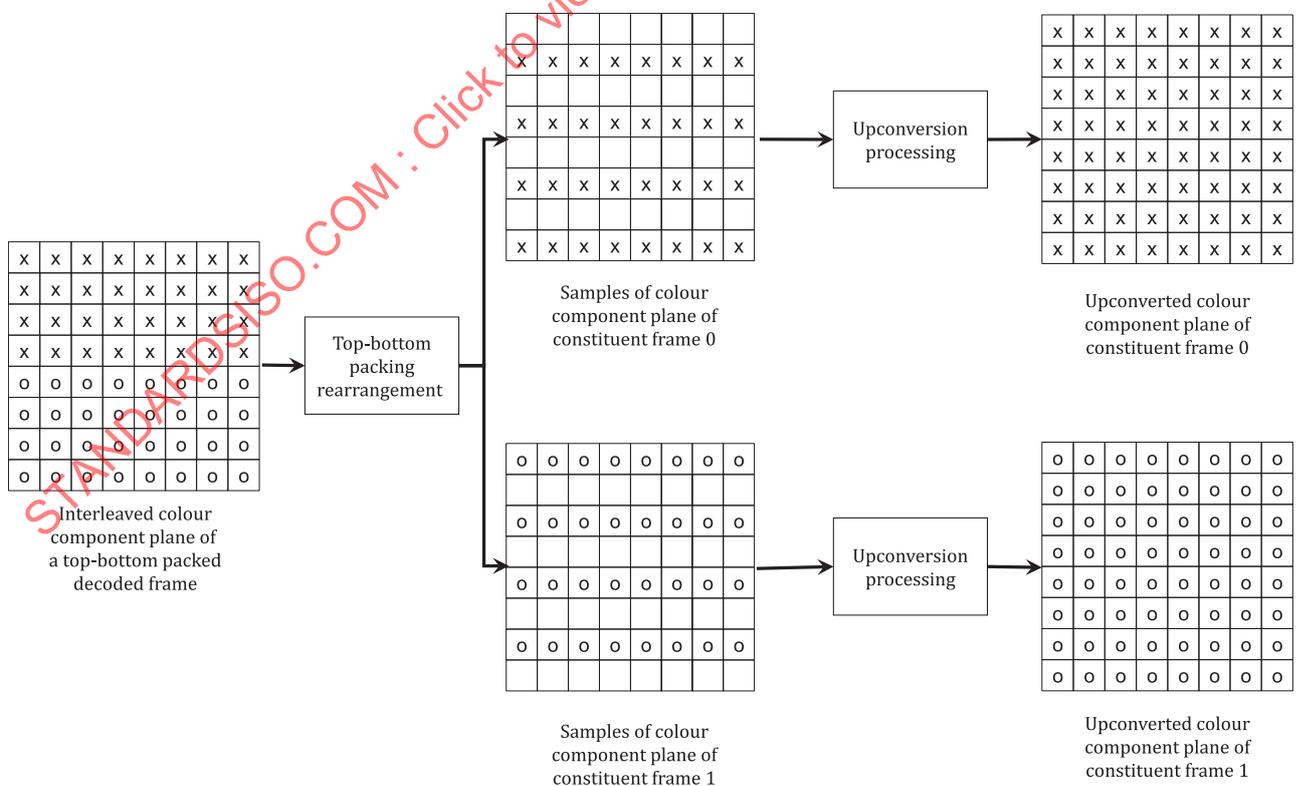


Figure 9 — Flowchart for rearrangement and upconversion of top-bottom packing arrangement with $fp_arrangement_type$ equal to 4, $fp_quincunx_sampling_flag$ equal to 0, (x, y) equal to $(8, 12)$ for constituent frame 0 and (x, y) equal to $(0, 0)$ or $(8, 4)$ for constituent frame 1

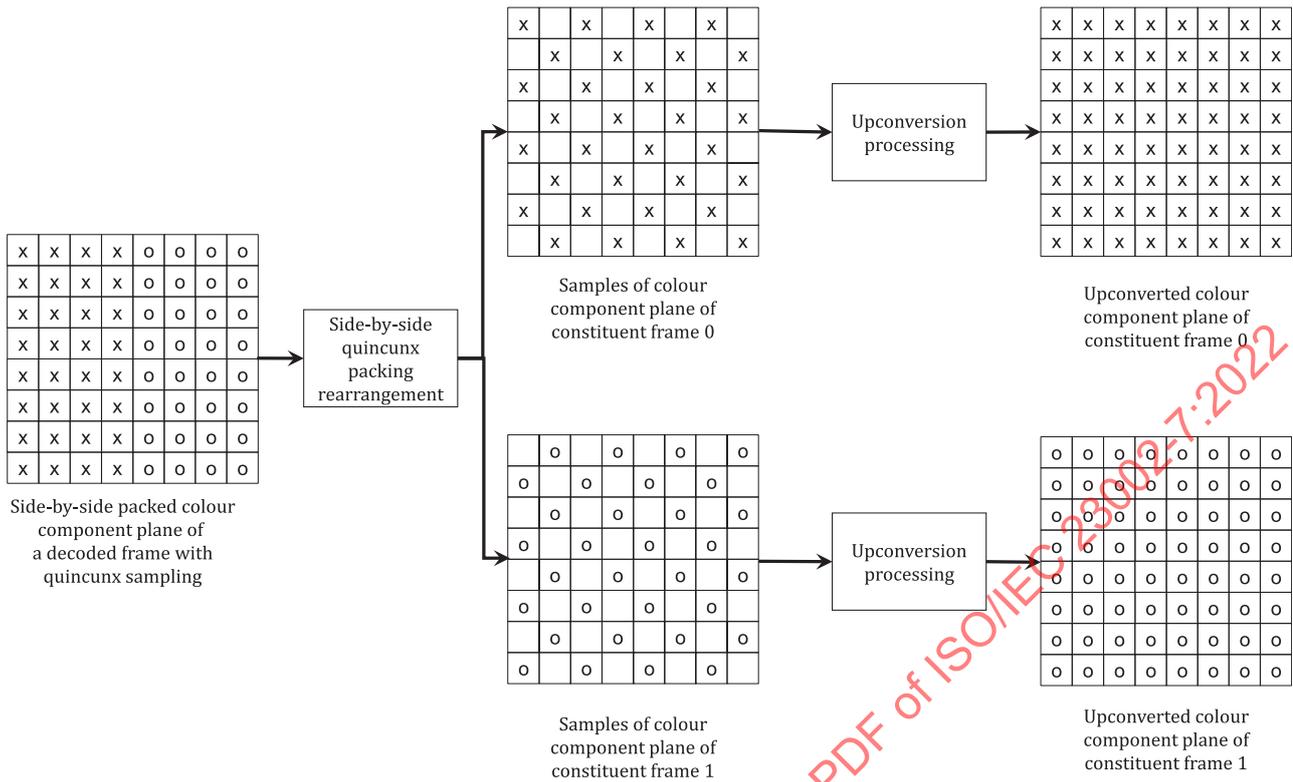


Figure 10 — Flowchart for rearrangement and upconversion of side-by-side packing arrangement with quincunx sampling (fp_arrangement_type equal to 3 with fp_quincunx_sampling_flag equal to 1)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23002-7:2022

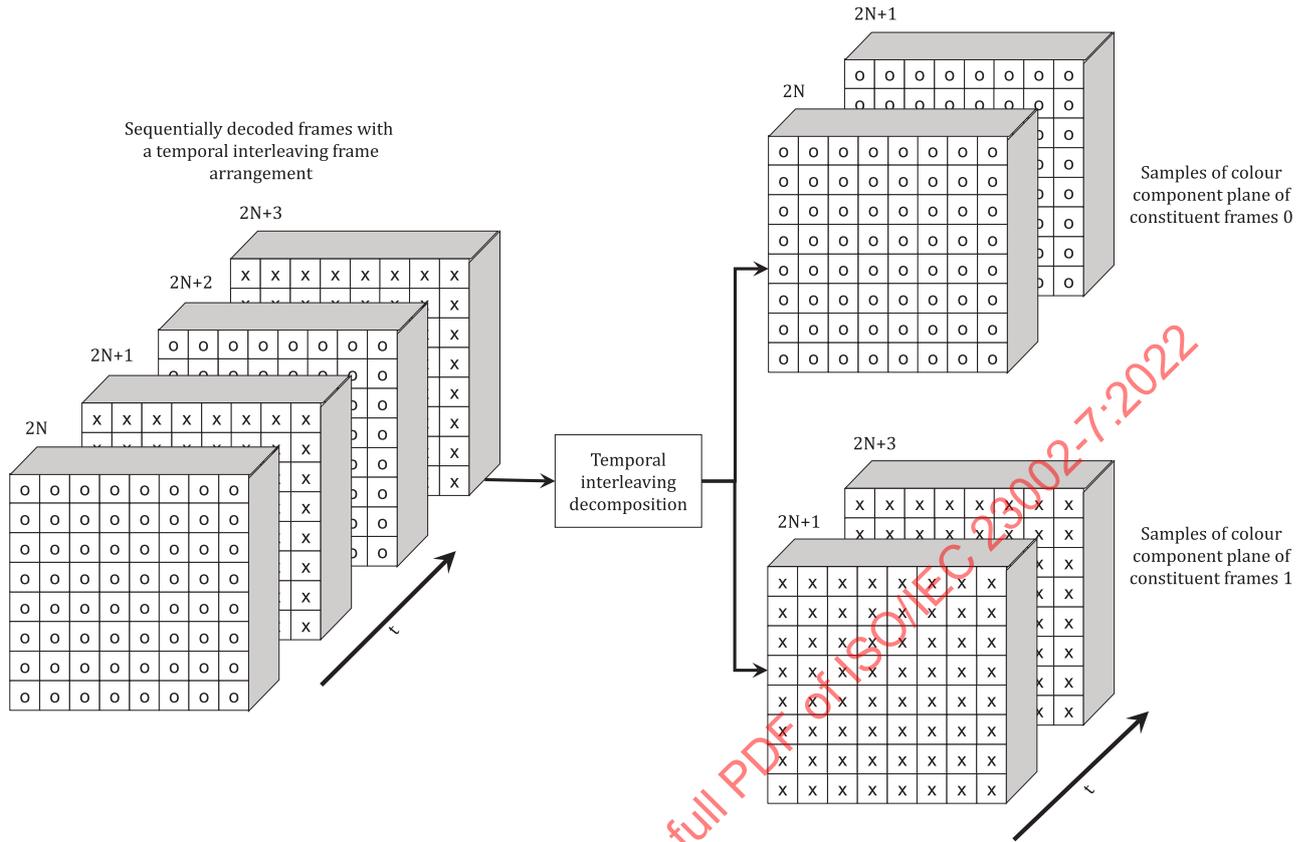


Figure 11 — Flowchart for rearrangement of a temporal interleaving frame arrangement (fp_arrangement_type equal to 5)

8.7 Parameter sets inclusion indication SEI message

8.7.1 Parameter sets inclusion indication SEI message syntax

parameter_sets_inclusion_indication(payloadSize) {	Descriptor
psii_self_contained_clvs_flag	u(1)
}	

8.7.2 Parameter sets inclusion indication SEI message semantics

This message provides an indication of whether the CLVS contains all the required NAL units for decoding the CLVS that is associated with the SEI message and whether temporal sublayer up-switching within the CLVS works without a need of fetching parameter sets from PUs earlier in decoding order than the PU containing the picture at which the temporal sublayer up-switching occurs. When the CLVS does not contain all the required NAL units, the NAL units that are not present in the CLVS may be provided externally.

psii_self_contained_clvs_flag equal to 1 indicates that the following restrictions apply:

- Each parameter set that is (directly or indirectly) referenced by any VCL NAL unit of the CLVS that is not a VCL NAL unit of a RASL picture (when present) associated with the first AU of the CLVS is present within the CLVS at a position that precedes, in decoding order, any NAL unit that (directly or indirectly) references the parameter set.

- For any STSA picture *stsaPicA* with temporal sublayer identifier equal to *tIdA* in the CLVS, the following applies:
 - *stsaPicA* does not refer to a PPS or an APS that precedes the first NAL unit of the PU containing *stsaPicA* in decoding order and has temporal sublayer identifier equal to *tIdA*.
 - For any picture *picB* with temporal sublayer identifier equal to *tIdA* and following *stsaPicA* in decoding order, *picB* does not refer to a PPS or an APS that has temporal sublayer identifier equal to *tIdA* that precedes the first NAL unit of the PU containing *stsaPicA* in decoding order.

psii_self_contained_clvs_flag equal to 0 indicates that this property might or might not apply.

8.8 Decoded picture hash SEI message

8.8.1 Decoded picture hash SEI message syntax

decoded_picture_hash(payloadSize) {	Descriptor
dph_sei_hash_type	u(8)
dph_sei_single_component_flag	u(1)
dph_sei_reserved_zero_7bits	u(7)
for(<i>cIdx</i> = 0; <i>cIdx</i> < (<i>dph_sei_single_component_flag</i> ? 1 : 3); <i>cIdx</i> ++)	
if(<i>dph_sei_hash_type</i> == 0)	
for(<i>i</i> = 0; <i>i</i> < 16; <i>i</i> ++)	
dph_sei_picture_md5[<i>cIdx</i>][<i>i</i>]	b(8)
else if(<i>dph_sei_hash_type</i> == 1)	
dph_sei_picture_crc[<i>cIdx</i>]	u(16)
else if(<i>dph_sei_hash_type</i> == 2)	
dph_sei_picture_checksum[<i>cIdx</i>]	u(32)
}	

8.8.2 Decoded picture hash SEI message semantics

This message provides a hash for each colour component of the current decoded picture.

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

- *Apicturewidthandpictureheightinunitsoflumasamples*, denoted herein by *PicWidthInLumaSamples* and *PicHeightInLumaSamples*, respectively.
- 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 *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*.
- For each colour component *cIdx*, an array of samples *ComponentSample[*cIdx*][*x*][*y*]*.

The variables *SubWidthC* and *SubHeightC* are derived from *ChromaFormatIdc* as specified by [Table 2](#).

Prior to computing the hash, the decoded picture data are arranged into one or three strings of bytes called *pictureData[*cIdx*]* of lengths *dataLen[*cIdx*]* as follows:

```

for( cIdx = 0; cIdx < dph_sei_single_component_flag ? 1 : 3; cIdx++ ) {
  if( cIdx == 0 ) {
    compWidth[ cIdx ] = PicWidthInLumaSamples
    compHeight[ cIdx ] = PicHeightInLumaSamples
    compDepth[ cIdx ] = BitDepthY
  } else {
    compWidth[ cIdx ] = PicWidthInLumaSamples / SubWidthC
    compHeight[ cIdx ] = PicHeightInLumaSamples / SubHeightC
    compDepth[ cIdx ] = BitDepthC
  }
  iLen = 0
  for( y = 0; y < compHeight[ cIdx ]; y++ ) /* raster scan order */
    for( x = 0; x < compWidth[ cIdx ]; x++ ) {
      pictureData[ cIdx ][ iLen++ ] = ComponentSample[ cIdx ][ x ][ y ] & 0xFF
      if( compDepth[ cIdx ] > 8 )
        pictureData[ cIdx ][ iLen++ ] = ComponentSample[ cIdx ][ x ][ y ] >> 8
    }
  dataLen[ cIdx ] = iLen
}

```

(29)

where `ComponentSample[cIdx]` is a 2-dimension array of the decoded sample values of a component of a decoded picture.

dph_sei_hash_type indicates the method used to calculate the checksum as specified in [Table 9](#). Values of `dph_sei_hash_type` that are not listed in [Table 9](#) are reserved for future use by ITU-T | ISO/IEC and shall not be present in payload data conforming to this edition of this document. Decoders shall ignore decoded picture hash SEI messages that contain reserved values of `dph_sei_hash_type`.

Table 9 — Interpretation of `dph_sei_hash_type`

<code>dph_sei_hash_type</code>	Method
0	MD5 (IETF RFC 1321)
1	CRC
2	Checksum

dph_sei_single_component_flag equal to 1 specifies that the picture associated with the decoded picture hash SEI message contains a single colour component. `dph_sei_single_component_flag` equal to 0 specifies that the picture associated with the decoded picture hash SEI message contains three colour components. The value of `dph_sei_single_component_flag` shall be equal to (`ChromaFormatIdc = 0`).

dph_sei_reserved_zero_7bits shall be equal to 0. Values greater than 0 for `dph_sei_reserved_zero_7bits` are reserved for future use by ITU-T | ISO/IEC and shall not be present in payload data conforming to this edition of this document. Decoders conforming to this edition of this document shall ignore the value of `dph_sei_reserved_zero_7bits`.

dph_sei_picture_md5[cIdx][i] is the 16-byte MD5 hash of the `cIdx`-th colour component of the decoded picture. The value of `dph_sei_picture_md5[cIdx][i]` shall be equal to the value of `digestVal[cIdx]` obtained as follows, using the MD5 functions defined in IETF RFC 1321:

```

MD5Init( context )
MD5Update( context, pictureData[ cIdx ], dataLen[ cIdx ] )
MD5Final( digestVal[ cIdx ], context )

```

(30)

dph_sei_picture_crc[cIdx] is the cyclic redundancy check (CRC) of the colour component `cIdx` of the decoded picture. The value of `dph_sei_picture_crc[cIdx]` shall be equal to the value of `crcVal[cIdx]` obtained as follows:

```

crc = 0xFFFF
pictureData[ cIdx ][ dataLen[ cIdx ] ] = 0
pictureData[ cIdx ][ dataLen[ cIdx ] + 1 ] = 0
for( bitIdx = 0; bitIdx < ( dataLen[ cIdx ] + 2 ) * 8; bitIdx++ ) {
    dataByte = pictureData[ cIdx ][ bitIdx >> 3 ]
    crcMsb = ( crc >> 15 ) & 1
    bitVal = ( dataByte >> ( 7 - ( bitIdx & 7 ) ) ) & 1
    crc = ( ( ( crc << 1 ) + bitVal ) & 0xFFFF ) ^ ( crcMsb * 0x1021 )
}
crcVal[ cIdx ] = crc
    
```

(31)

NOTE The same CRC specification is found in Rec. ITU-T H.271.

dph_sei_picture_checksum[cIdx] is the checksum of the colour component cIdx of the decoded picture. The value of dph_sei_picture_checksum[cIdx] shall be equal to the value of checksumVal[cIdx] obtained as follows:

```

sum = 0
for( y = 0; y < compHeight[ cIdx ]; y++ )
    for( x = 0; x < compWidth[ cIdx ]; x++ ) {
        xorMask = ( x & 0xFF ) ^ ( y & 0xFF ) ^ ( x >> 8 ) ^ ( y >> 8 )
        sum = ( sum + ( ( ComponentSample[ cIdx ][ y * compWidth[ cIdx ] + x ] & 0xFF ) ^
            xorMask ) ) & 0xFFFFFFFF
        if( compDepth[ cIdx ] > 8 )
            sum = ( sum + ( ( ComponentSample[ cIdx ][ y * compWidth[ cIdx ] + x ] >> 8 ) ^
                xorMask ) ) & 0xFFFFFFFF
    }
checksumVal[ cIdx ] = sum
    
```

(32)

8.9 Mastering display colour volume SEI message

8.9.1 Mastering display colour volume SEI message syntax

mastering_display_colour_volume(payloadSize) {	Descriptor
for(c = 0; c < 3; c++) {	
mdcv_display primaries_x [c]	u(16)
mdcv_display primaries_y [c]	u(16)
}	
mdcv_white_point_x	u(16)
mdcv_white_point_y	u(16)
mdcv_max_display_mastering_luminance	u(32)
mdcv_min_display_mastering_luminance	u(32)
}	

8.9.2 Mastering display colour volume SEI message semantics

This SEI message identifies the colour volume (the colour primaries, white point, and luminance range) of a display considered to be the mastering display for the associated video content e.g., the colour volume of a display that was used for viewing while authoring the video content. The described mastering display is a three-colour additive display system that has been configured to use the indicated mastering colour volume.

This SEI message does not identify the measurement methodologies and procedures used for determining the indicated values or provide any description of the mastering environment. It also does

not provide information on colour transformations that would be appropriate to preserve creative intent on displays with colour volumes different from that of the described mastering display.

The information conveyed in this SEI message is intended to be adequate for purposes corresponding to the use of SMPTE ST 2086.

When a mastering display colour volume SEI message is present for any picture of a CLVS of a particular layer, a mastering display colour volume SEI message shall be present for the first picture of the CLVS. The mastering display colour volume SEI message persists for the current layer in decoding order from the current picture until the end of the CLVS. All mastering display colour volume SEI messages that apply to the same CLVS shall have the same content.

mdcv_display primaries_x[c], when in the range of 5 to 37 000, inclusive, specifies the normalized x chromaticity coordinate of the colour primary component c of the mastering display, according to the CIE 1931 definition of x as specified in ISO/CIE 11664-1 (see also ISO/CIE 11664-3 and CIE 15), in increments of 0.00002. When **mdcv_display primaries_x[c]** is not in the range of 5 to 37 000, inclusive, the normalized x chromaticity coordinate of the colour primary component c of the mastering display is unknown or unspecified or specified by other means not specified in this document.

mdcv_display primaries_y[c], when in the range of 5 to 42 000, inclusive, specifies the normalized y chromaticity coordinate of the colour primary component c of the mastering display, according to the CIE 1931 definition of y as specified in ISO/CIE 11664-1 (see also ISO/CIE 11664-3 and CIE 15), in increments of 0.00002. When **mdcv_display primaries_y[c]** is not in the range of 5 to 42 000, inclusive, the normalized y chromaticity coordinate of the colour primary component c of the mastering display is unknown or unspecified or specified by other means not specified in this document.

For describing mastering displays that use red, green, and blue colour primaries, it is suggested that index value c equal to 0 should correspond to the green primary, c equal to 1 should correspond to the blue primary, and c equal to 2 should correspond to the red colour primary specified in the VUI parameters.

mdcv_white_point_x, when in the range of 5 to 37 000, inclusive, specifies the normalized x chromaticity coordinate of the white point of the mastering display, according to the CIE 1931 definition of x as specified in ISO/CIE 11664-1 (see also ISO/CIE 11664-3 and CIE 15), in normalized increments of 0.00002. When **mdcv_white_point_x** is not in the range of 5 to 37 000, inclusive, the normalized x chromaticity coordinate of the white point of the mastering display is indicated to be unknown or unspecified or specified by other means not specified in this document.

mdcv_white_point_y, when in the range of 5 to 42 000, inclusive, specifies the normalized y chromaticity coordinate of the white point of the mastering display, according to the CIE 1931 definition of y as specified in ISO/CIE 11664-1 (see also ISO/CIE 11664-3 and CIE 15), in normalized increments of 0.00002. When **mdcv_white_point_y** is not in the range of 5 to 42 000, inclusive, the normalized y chromaticity coordinate of the white point of the mastering display is indicated to be unknown or unspecified or specified by other means not specified in this document.

NOTE 1 SMPTE ST 2086 specifies that the normalized x and y chromaticity coordinate values for the mastering display colour primaries and white point are to be represented with four decimal places. This would correspond with using values of the syntax elements **mdcv_display primaries_x[c]**, **mdcv_display primaries_y[c]**, **mdcv_white_point_x**, and **mdcv_white_point_y**, as defined in this document, that are multiples of 5.

NOTE 2 An example of the use of values outside the range for which semantics are specified in this document is that ANSI/CTA 861-G uses normalized (x, y) chromaticity coordinate values of (0,0) for the white point to indicate that the white point chromaticity is unknown.

mdcv_max_display mastering luminance, when in the range of 50 000 to 100 000 000, specifies the nominal maximum display luminance of the mastering display in units of 0.0001 candelas per square metre. When **mdcv_max_display mastering luminance** is not in the range of 50 000 to 100 000 000,

the nominal maximum display luminance of the mastering display is indicated to be unknown or unspecified or specified by other means not specified in this document.

NOTE 3 SMPTE ST 2086 specifies that the nominal maximum display luminance of the mastering display is to be specified as a multiple of 1 candela per square metre. This would correspond with using values of the syntax element `mdcv_max_display_mastering_luminance`, as defined in this document, that are a multiple of 10 000.

NOTE 4 An example of the use of values outside the range for which semantics are specified in this document is that ANSI/CTA 861-G uses the value 0 for the nominal maximum display luminance of the mastering display to indicate that the nominal maximum display luminance of the mastering display is unknown.

mdcv_min_display_mastering_luminance, when in the range of 1 to 50 000, specifies the nominal minimum display luminance of the mastering display in units of 0.0001 candelas per square metre. When `mdcv_min_display_mastering_luminance` is not in the range of 1 to 50 000, the nominal maximum display luminance of the mastering display is unknown or unspecified or specified by other means not specified in this document. When `mdcv_max_display_mastering_luminance` is equal to 50 000, `mdcv_min_display_mastering_luminance` shall not be equal to 50 000.

NOTE 5 SMPTE ST 2086 specifies that the nominal minimum display luminance of the mastering display is to be specified as a multiple of 0.0001 candelas per square metre, which corresponds to the semantics specified in this document.

NOTE 6 An example of the use of values outside the range for which semantics are specified in this document is that ANSI/CTA 861-G uses the value 0 for the nominal minimum display luminance of the mastering display to indicate that the nominal minimum display luminance of the mastering display is unknown.

NOTE 7 Another example of the potential use of values outside the range for which semantics are specified in this document is that SMPTE ST 2086 indicates that values outside the specified range could be used to indicate that the black level and contrast of the mastering display have been adjusted using picture line-up generation equipment (PLUGE).

At the minimum luminance, the mastering display is considered to have the same nominal chromaticity as the white point.

8.10 Content light level information SEI message

8.10.1 Content light level information SEI message syntax

	Descriptor
<code>content_light_level_info(payloadSize) {</code>	
<code>clli_max_content_light_level</code>	u(16)
<code>clli_max_pic_average_light_level</code>	u(16)
<code>}</code>	

8.10.2 Content light level information SEI message semantics

This SEI message identifies upper bounds for the nominal target brightness light level of the pictures of the CLVS.

The information conveyed in this SEI message is intended to be adequate for purposes corresponding to the use of CEA 861.3.

The semantics of the content light level information SEI message are defined in relation to the values of samples in a 4:4:4 representation of red, green, and blue colour primary intensities in the linear light domain for the pictures of the CLVS, in units of candelas per square metre. However, this SEI message does not, by itself, identify a conversion process for converting the sample values of a decoded picture to the samples in a 4:4:4 representation of red, green, and blue colour primary intensities in the linear light domain for the picture.

NOTE 1 Other syntax elements, such as `vui_colour primaries`, `vui_transfer_characteristics`, and `vui_matrix_coeffs`, when present, could assist in the identification of such a conversion process.

Given the red, green, and blue colour primary intensities in the linear light domain for the location of a luma sample in a corresponding 4:4:4 representation, denoted as E_R , E_G , and E_B , the maximum component intensity is defined as $E_{Max} = \text{Max}(E_R, \text{Max}(E_G, E_B))$. The light level corresponding to the stimulus is then defined as the CIE 1931 luminance corresponding to equal amplitudes of E_{Max} for all three colour primary intensities for red, green, and blue (with appropriate scaling to reflect the nominal luminance level associated with peak white – e.g., ordinarily scaling to associate peak white with 10 000 candelas per square metre when `vui_transfer_characteristics` is equal to 16).

NOTE 2 Since the maximum value E_{Max} is used in this definition at each sample location, rather than a direct conversion from E_R , E_G , and E_B to the corresponding CIE 1931 luminance, the CIE 1931 luminance at a location could in some cases be less than the indicated light level. This situation would occur, for example, when E_R and E_G are very small and E_B is large, in which case the indicated light level would be much larger than the true CIE 1931 luminance associated with the (E_R, E_G, E_B) triplet.

All content light level information SEI messages that apply to the same CLVS shall have the same content.

cli_max_content_light_level, when not equal to 0, indicates an upper bound on the maximum light level among all individual samples in a 4:4:4 representation of red, green, and blue colour primary intensities (in the linear light domain) for the pictures of the CLVS, in units of candelas per square metre. When equal to 0, no such upper bound is indicated by `cli_max_content_light_level`.

cli_max_pic_average_light_level, when not equal to 0, indicates an upper bound on the maximum average light level among the samples in a 4:4:4 representation of red, green, and blue colour primary intensities (in the linear light domain) for any individual picture of the CLVS, in units of candelas per square metre. When equal to 0, no such upper bound is indicated by `cli_max_pic_average_light_level`.

When the visually relevant region does not correspond to the entire cropped decoded picture, such as for "letterbox" encoding of video content with a wide picture aspect ratio within a taller cropped decoded picture, the indicated average should be performed only within the visually relevant region.

8.11 Dependent random access point indication SEI message

8.11.1 Dependent random access point indication SEI message syntax

<code>dependent_rap_indication(payloadSize) {</code>	Descriptor
<code>}</code>	

8.11.2 Dependent random access point indication SEI message semantics

The picture associated with a dependent random access point (DRAP) indication SEI message is referred to as a DRAP picture.

The presence of the DRAP indication SEI message indicates that the constraints on picture order and picture referencing specified in this subclause apply. These constraints can enable a decoder to properly decode the DRAP picture and the pictures that are in the same layer and follow it in both decoding order and output order without needing to decode any other pictures in the same layer except the associated IRAP picture of the DRAP picture.

The constraints indicated by the presence of the DRAP indication SEI message, which shall all apply, are as follows:

- The DRAP picture is a trailing picture.
- The DRAP picture has a temporal sublayer identifier equal to 0.
- The DRAP picture does not include any pictures in the same layer in the active entries of its reference picture lists except the associated IRAP picture of the DRAP picture.
- Any picture that is in the same layer and follows the DRAP picture in both decoding order and output order does not include, in the active entries of its reference picture lists, any picture that is in

the same layer and precedes the DRAP picture in decoding order or output order, with the exception of the associated IRAP picture of the DRAP picture.

8.12 Alternative transfer characteristics information SEI message

8.12.1 Alternative transfer characteristics information SEI message syntax

alternative_transfer_characteristics (payloadSize) {	Descriptor
preferred_transfer_characteristics	u(8)
}	

8.12.2 Alternative transfer characteristics SEI message semantics

The alternative transfer characteristics SEI message provides a preferred alternative value for the transfer_characteristics syntax element that is indicated by the colour description syntax of the VUI parameters. This SEI message is intended to be used in cases when some value of vui_transfer_characteristics is preferred for interpretation of the pictures of the CLVS although some other value of vui_transfer_characteristics could also be acceptable for interpretation of the pictures of the CLVS and that other value is provided in the colour description syntax of the VUI parameters for interpretation by decoders that do not support interpretation of the preferred value (e.g., because the preferred value had not yet been defined in a previous edition of this document).

When an alternative transfer characteristics SEI message is present for any picture of a CLVS of a particular layer and the first picture of the CLVS is an IRAP picture, an alternative transfer characteristics SEI message shall be present for that IRAP picture. The alternative transfer characteristics SEI message persists for the current layer in decoding order from the current picture until the end of the CLVS. All alternative transfer characteristics SEI messages that apply to the same CLVS shall have the same content.

preferred_transfer_characteristics specifies a preferred alternative value for the vui_transfer_characteristics syntax element of the colour description syntax of the VUI parameters. The semantics for preferred_transfer_characteristics are otherwise the same as for the vui_transfer_characteristics syntax element specified in the VUI parameters. When preferred_transfer_characteristics is not equal to the value of vui_transfer_characteristics indicated in the VUI parameters, decoders should ignore the value of vui_transfer_characteristics indicated in the VUI parameters and instead use the value indicated by preferred_transfer_characteristics.

8.13 Ambient viewing environment SEI message

8.13.1 Ambient viewing environment SEI message syntax

ambient_viewing_environment(payloadSize) {	Descriptor
ambient_illuminance	u(32)
ambient_light_x	u(16)
ambient_light_y	u(16)
}	

8.13.2 Ambient viewing environment SEI message semantics

The ambient viewing environment SEI message identifies the characteristics of the nominal ambient viewing environment for the display of the associated video content. The syntax elements of the ambient viewing environment SEI message can assist the receiving system in adapting the received video content for local display in viewing environments that could be similar or could substantially differ from those assumed or intended when mastering the video content.

This SEI message does not provide information on colour transformations that would be appropriate to preserve creative intent on displays with colour volumes different from that of the described mastering display.

When an ambient viewing environment SEI message is present for any picture of a CLVS of a particular layer and the first picture of the CLVS is an IRAP picture, an ambient viewing environment SEI message shall be present for that IRAP picture. The ambient viewing environment SEI message persists for the current layer in decoding order from the current picture until the end of the CLVS. All ambient viewing environment SEI messages that apply to the same CLVS shall have the same content.

ambient_illuminance specifies the environmental illuminance of the ambient viewing environment in units of 0.0001 lux. **ambient_illuminance** shall not be equal to 0.

ambient_light_x and **ambient_light_y** specify the normalized x and y chromaticity coordinates, respectively, of the environmental ambient light in the nominal viewing environment, according to the CIE 1931 definition of x and y as specified in ISO/CIE 11664-1 (see also ISO/CIE 11664-3 and CIE 15), in normalized increments of 0.00002. The values of **ambient_light_x** and **ambient_light_y** shall be in the range of 0 to 50 000.

NOTE For example, the conditions identified in Rec. ITU-R BT.2035 could be expressed using **ambient_illuminance** equal to 100 000 with background chromaticity indicating D_{65} (**ambient_light_x** equal to 15 635, **ambient_light_y** equal to 16 450), or optionally in some regions, background chromaticity indicating D_{93} (**ambient_light_x** equal to 14 155, **ambient_light_y** equal to 14 855).

8.14 Content colour volume SEI message

8.14.1 Content colour volume SEI message syntax

content_colour_volume(payloadSize) {	Descriptor
ccv_cancel_flag	u(1)
if(!ccv_cancel_flag) {	
ccv_persistence_flag	u(1)
ccv primaries_present_flag	u(1)
ccv_min_luminance_value_present_flag	u(1)
ccv_max_luminance_value_present_flag	u(1)
ccv_avg_luminance_value_present_flag	u(1)
ccv_reserved_zero_2bits	u(2)
if(ccv_primaries_present_flag)	
for(c = 0; c < 3; c++) {	
ccv_primaries_x[c]	i(32)
ccv_primaries_y[c]	i(32)
}	
if(ccv_min_luminance_value_present_flag)	
ccv_min_luminance_value	u(32)
if(ccv_max_luminance_value_present_flag)	
ccv_max_luminance_value	u(32)
if(ccv_avg_luminance_value_present_flag)	
ccv_avg_luminance_value	u(32)
}	
}	

8.14.2 Content colour volume SEI message semantics

The content colour volume SEI message describes the colour volume characteristics of the associated pictures. These colour volume characteristics are expressed in terms of a nominal range, although deviations from this range may occur.

The variable `transferCharacteristics` is specified as follows:

- If an alternative transfer characteristics SEI message is present for the CLVS, `transferCharacteristics` is set equal to `preferred_transfer_characteristics`;
- Otherwise, (an alternative transfer characteristics SEI message is not present for the CLVS), `transferCharacteristics` is set equal to `vui_transfer_characteristics`.

The content colour volume SEI message shall not be present, and decoders shall ignore it, when any of the following conditions is true:

- Any of the values of `transferCharacteristics`, `vui_colour_primaries`, and `vui_matrix_coeffs` has a value defined as unknown or unspecified.
- The value of `vui_transfer_characteristics` is equal to 2, 4, or 5.
- The value of `vui_colour_primaries` is equal to 2.

The following applies when converting the signal from a non-linear to a linear representation:

- If the value of `transferCharacteristics` is equal to 1, 6, 7, 14, or 15, the Rec. ITU-R BT.1886 reference electro-optical transfer function should be used to convert the signal to its linear representation, where the value of screen luminance for white is set equal to 100 candelas per square metre, the value of screen luminance for black is set equal to 0 candelas per square metre, and the value of the exponent of the power function is set equal to 2.4.
- Otherwise, if the value of `transferCharacteristics` is equal to 18, the hybrid log-gamma reference electro-optical transfer function specified in Rec. ITU-R BT.2100 should be used to convert the signal to its linear representation, where the value of nominal peak luminance of the display is set equal to 1000 candelas per square metre, the value of the display luminance for black is set equal to 0 candelas per square metre, and the value of system gamma is set equal to 1.2.
- Otherwise (the value of `transferCharacteristics` is not equal to 1, 6, 7, 14, 15, or 18) when the content colour volume SEI message is present, the exact inverse of the transfer function specified in specified in the VUI parameters should be used to convert the non-linear signal to a linear representation.

`ccv_cancel_flag` equal to 1 indicates that the SEI message cancels the persistence of any previous content colour volume SEI message in output order that applies to the current layer. `ccv_cancel_flag` equal to 0 indicates that content colour volume information follows.

`ccv_persistence_flag` specifies the persistence of the content colour volume SEI message for the current layer.

`ccv_persistence_flag` equal to 0 specifies that the content colour volume applies to the current decoded picture only.

`ccv_persistence_flag` equal to 1 specifies that the content colour volume 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 content colour volume SEI message is output that follows the current picture in output order.

ccv_primaries_present_flag equal to 1 specifies that the syntax elements `ccv_primaries_x[c]` and `ccv_primaries_y[c]` are present. `ccv_primaries_present_flag` equal to 0 specifies that the syntax elements `ccv_primaries_x[c]` and `ccv_primaries_y[c]` are not present.

ccv_min_luminance_value_present_flag equal to 1 specifies that the syntax element `ccv_min_luminance_value` is present. `ccv_min_luminance_value_present_flag` equal to 0 specifies that the syntax element `ccv_min_luminance_value` is not present.

ccv_max_luminance_value_present_flag equal to 1 specifies that the syntax element `ccv_max_luminance_value` is present. `ccv_max_luminance_value_present_flag` equal to 0 specifies that the syntax element `ccv_max_luminance_value` is not present.

ccv_avg_luminance_value_present_flag equal to 1 specifies that the syntax element `ccv_avg_luminance_value` is present. `ccv_avg_luminance_value_present_flag` equal to 0 specifies that the syntax element `ccv_avg_luminance_value` is not present.

It is a requirement of bitstream conformance that the values of `ccv_primaries_present_flag`, `ccv_min_luminance_value_present_flag`, `ccv_max_luminance_value_present_flag`, and `ccv_avg_luminance_value_present_flag` shall not all be equal to 0.

ccv_reserved_zero_2bits[i] shall be equal to 0 in bitstreams conforming to this edition of this document. Other values for `reserved_zero_2bits[i]` are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of `reserved_zero_2bits[i]`.

ccv_primaries_x[c] and **ccv_primaries_y[c]** specify the normalized x and y chromaticity coordinates, respectively, of the colour primary component c of the nominal content colour volume, according to the CIE 1931 definition of x and y as specified in ISO/CIE 11664-1 (see also ISO/CIE 11664-3 and CIE 15), in normalized increments of 0.00002. For describing colour volumes that use red, green, and blue colour primaries, it is suggested that index value c equal to 0 should correspond to the green primary, c equal to 1 should correspond to the blue primary, and c equal to 2 should correspond to the red colour primary specified in the VUI parameters.

The values of `ccv_primaries_x[c]` and `ccv_primaries_y[c]` shall be in the range of -5 000 000 to 5 000 000, inclusive.

When `ccv_primaries_x[c]` and `ccv_primaries_y[c]` are not present, they are inferred to be equal to the normalized x and y chromaticity coordinates, respectively, specified by `vui_colour_primaries`.

ccv_min_luminance_value specifies the normalized minimum luminance value, according to CIE 1931, that is expected to be present in the content, where values are normalized to L_0 or L_c as specified in the VUI parameters according to the indicated transfer characteristics of the signal. The values of `ccv_min_luminance_value` are in normalized increments of 0.0000001.

ccv_max_luminance_value specifies the maximum luminance value, according to CIE 1931, that is expected to be present in the content, where values are normalized to L_0 or L_c as specified in the VUI parameters according to the transfer characteristics of the signal. The values of `ccv_max_luminance_value` are in normalized increments of 0.0000001.

ccv_avg_luminance_value specifies the average luminance value, according to CIE 1931, that is expected to be present in the content, where values are normalized to L_0 or L_c as specified in the VUI parameters according to the transfer characteristics of the signal. The values of `ccv_avg_luminance_value` are in normalized increments of 0.0000001.

NOTE The resulting domain from this conversion process might or might not represent light in a source or display domain – it is merely a gamut representation domain rather than necessarily being a representation of actual light in either the scene or display domain. Therefore, the values corresponding to `ccv_min_luminance_value`, `ccv_max_luminance_value`, and `ccv_avg_luminance_value` might not necessarily correspond to a true luminance value.

The value of `ccv_min_luminance_value`, when present, shall be less than or equal to `ccv_avg_luminance_value`, when present. The value of `ccv_avg_luminance_value`, when present, shall be less than or equal to

ccv_max_luminance_value, when present. The value of ccv_min_luminance_value, when present, shall be less than or equal to ccv_max_luminance_value, when present.

When the visually relevant region does not correspond to the entire cropped decoded picture, such as for "letterbox" encoding of video content with a wide picture aspect ratio within a taller cropped decoded picture, the indicated ccv_min_luminance_value, ccv_max_luminance_value, and ccv_avg_luminance_value should correspond only to values within the visually relevant region.

8.15 Omnidirectional video specific SEI messages

8.15.1 Sample location remapping process

8.15.1.1 General

Use of this process requires the definition of the following variable:

- A chroma format indicator, denoted herein by ChromaFormatIdc, as described in [subclause 7.3](#).

To remap colour sample locations of a region-wise packed picture to a unit sphere, the following ordered steps are applied:

1. A region-wise packed picture is obtained as the cropped decoded picture by decoding a coded picture. For purposes of interpretation of chroma samples, the input to the indicated remapping process is the set of decoded sample values after applying an (unspecified) upsampling conversion process to the 4:4:4 colour sampling format as necessary when ChromaFormatIdc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format). This (unspecified) upsampling process should account for the relative positioning relationship between the luma and chroma samples as indicated by vui_chroma_sample_loc_type_frame, vui_chroma_sample_loc_type_top_field, and vui_chroma_sample_loc_type_bottom_field syntax elements in the VUI parameters, when present.
2. If RWP is indicated, the sample locations of the region-wise packed picture are converted to sample locations of the respective projected picture as specified in [subclause 8.15.1.4](#). Otherwise, the projected picture is identical to the region-wise packed picture.
3. If frame packing is indicated, the sample locations of the projected picture are converted to sample locations of the respective constituent picture of the projected picture, as specified in [subclause 8.15.1.5](#). Otherwise, the constituent picture of the projected picture is identical to the projected picture.
4. The sample locations of a constituent picture of the projected picture are converted to sphere coordinates relative to the local coordinate axes, as specified in [subclause 8.15.1.2](#).
5. If rotation is indicated, the sphere coordinates relative to the local coordinate axes are converted to sphere coordinates relative to the global coordinate axes, as specified in [subclause 8.15.1.3](#). Otherwise, the global coordinate axes are identical to the local coordinate axes.

The overall process for mapping of luma sample locations within a region-wise packed picture to sphere coordinates relative to the global coordinate axes is normatively specified in [subclause 8.15.1.5](#).

For each region-wise packed picture corresponding to a decoded picture, the following applies:

- When an equirectangular projection SEI message with erp_cancel_flag equal to 0 that applies to the picture is present, ErpFlag is set equal to 1, and CmpFlag is set equal to 0.
- When a generalized cubemap projection SEI message with gcmp_cancel_flag equal to 0 that applies to the picture is present, CmpFlag is set equal to 1, and ErpFlag is set equal to 0.
- If a sphere rotation SEI message with sphere_rotation_cancel_flag equal to 0 that applies to the picture is present, RotationFlag is set equal to 1, and RotationYaw, RotationPitch, and RotationRoll are set equal to yaw_rotation $\div 2^{16}$, pitch_rotation $\div 2^{16}$, and roll_rotation $\div 2^{16}$, respectively.

- Otherwise, RotationFlag is set equal to 0.
- If a frame packing arrangement SEI message with fp_arrangement_cancel_flag equal to 0 that applies to the picture is not present, StereoFlag, TopBottomFlag, and SideBySideFlag are all set equal to 0, HorDiv1 is set equal to 1, and VerDiv1 is set equal to 1.
- Otherwise, the following applies:
 - StereoFlag is set equal to 1.
 - If the value of fp_arrangement_type of the frame packing arrangement SEI message is equal to 3, TopBottomFlag is set equal to 0, SideBySideFlag is set equal to 1, HorDiv1 is set equal to 2 and VerDiv1 is set equal to 1.
 - Otherwise, if the value of fp_arrangement_type of the frame packing arrangement SEI message is equal to 4, TopBottomFlag is set equal to 1, SideBySideFlag is set equal to 0, HorDiv1 is set equal to 1, and VerDiv1 is set equal to 2.
 - Otherwise, TopBottomFlag is set equal to 0, SideBySideFlag is set equal to 0, HorDiv1 is set equal to 1, and VerDiv1 is set equal to 1.
- If a RWP SEI message with rwp_cancel_flag equal to 0 that applies to the picture is not present, RegionWisePackingFlag is set equal to 0, and ConstituentPicWidth and ConstituentPicHeight are set to be equal to cropPicWidth / HorDiv1 and cropPicHeight / VerDiv1, respectively, where cropPicWidth and cropPicHeight are the width and height, respectively, of the cropped decoded picture.
- Otherwise, RegionWisePackingFlag is set equal to 1, and ConstituentPicWidth and ConstituentPicHeight are set equal to rwp_proj_picture_width / HorDiv1 and rwp_proj_picture_height / VerDiv1, respectively.

8.15.1.2 Projection for one sample location

Inputs to this process are:

- pictureWidth and pictureHeight, which are the width and height, respectively, of a monoscopic projected luma picture, in relative projected picture sample units (see [subclause 8.15.5.2](#)), and
- the centre point of a sample location (hPos, vPos) along the horizontal and vertical axes, respectively, in relative projected picture sample units, where hPos and vPos could have non-integer real values.

Outputs of this process are:

- sphere coordinates (ϕ , θ) for the sample location in degrees relative to the local coordinate axes.

The projection for a sample location is derived as follows:

- If ErpFlag is equal to 1, the following applies:

- If RegionWisePackingFlag is equal to 0 and erp_guard_band_flag is equal to 1, the following applies:

$$\begin{aligned} hPos' &= hPos - erp_left_guard_band_width \\ pictureWidth &= pictureWidth - erp_left_guard_band_width - erp_right_guard_band_width \end{aligned} \quad (33)$$

- Otherwise, the following applies:

$$hPos' = hPos \quad (34)$$

- The following applies:

$$\begin{aligned} \phi &= 180 - hPos' * (360 \div pictureWidth) \\ \theta &= 90 - vPos * (180 \div pictureHeight) \end{aligned} \quad (35)$$

- Otherwise (CmpFlag is equal to 1), the outputs are derived by the following ordered steps:
 1. [Subclause 8.15.1.7](#) is invoked with pictureWidth and pictureHeight as inputs, and the output is assigned to faceWidth and faceHeight.
 2. [Subclause 8.15.1.8](#) is invoked with hPos, vPos, faceWidth, and faceHeight, where hPos and vPos are within a projected picture, and the output is assigned to hPosFace and vPosFace within a projected face.
 3. [Subclause 8.15.1.9](#) is invoked with hPosFace, vPosFace, faceWidth, and faceHeight, and the output is assigned to hPosRot and vPosRot.
 4. If gcmp_packing_type is equal to 4 or 5, [subclause 8.15.1.10](#) is invoked with hPosRot, vPosRot, faceWidth, and faceHeight, and the output is assigned to hPosAdj and vPosAdj. Otherwise, hPosAdj and vPosAdj are identical to hPosRot and vPosRot, respectively.
 5. The following applies:

$$\begin{aligned} hPos' &= - (2 * hPosAdj \div faceWidth) + 1 \\ vPos' &= - (2 * vPosAdj \div faceHeight) + 1 \end{aligned} \tag{36}$$

- If gcmp_mapping_function_type is equal to 0, the following applies:

$$\begin{aligned} hPos'' &= hPos' \\ vPos'' &= vPos' \end{aligned} \tag{37}$$

- Otherwise, if gcmp_mapping_function_type is equal to 1, the following applies:

$$\begin{aligned} hPos'' &= \text{Tan}(hPos' * \pi \div 4) \\ vPos'' &= \text{Tan}(vPos' * \pi \div 4) \end{aligned} \tag{38}$$

- Otherwise (gcmp_mapping_function_type is equal to 2), the following applies:

$$\begin{aligned} \text{coeffU}[n] &= (\text{gcmp_function_coeff_u}[n] + 1) \div 128 \\ \text{coeffV}[n] &= (\text{gcmp_function_coeff_v}[n] + 1) \div 128 \\ hPos'' &= hPos' \div (1 + \text{coeffU}[n] * (1 - \text{gcmp_function_u_affected_by_v_flag}[n] * vPos'^2) * \\ & \quad (1 - hPos'^2)) \\ vPos'' &= vPos' \div (1 + \text{coeffV}[n] * (1 - \text{gcmp_function_v_affected_by_u_flag}[n] * hPos'^2) * \\ & \quad (1 - vPos'^2)) \end{aligned} \tag{39}$$

- The following applies:

```

if( gcmp_face_index[ n ] == 0 ) { /* positive x front face */
  x = 1.0
  y = hPos''
  z = vPos''
} else if( gcmp_face_index[ n ] == 1 ) { /* negative x back face */
  x = -1.0
  y = -vPos''
  z = -hPos''
} else if( gcmp_face_index[ n ] == 2 ) { /* positive z top face */
  x = -hPos''
  y = -vPos''
  z = 1.0
} else if( gcmp_face_index[ n ] == 3 ) { /* negative z bottom face */
  x = hPos''
  y = -vPos''
  z = -1.0
} else if( gcmp_face_index[ n ] == 5 ) { /* positive y left face */
  x = -hPos''
  y = 1.0
  z = vPos''
} else { /* ( gcmp_face_index[ n ] == 4 ), negative y right face */
  x = hPos''
  y = -1.0
  z = vPos''
}
}
 $\phi = \text{Atan2}( y, x ) * 180 \div \pi$ 
 $\theta = \text{Asin}( z \div \text{Sqrt}( x^2 + y^2 + z^2 ) ) * 180 \div \pi$ 

```

(40)

8.15.1.3 Conversion from the local coordinate axes to the global coordinate axes

Inputs to this process are:

- rotation_yaw (α_d), rotation_pitch (β_d), rotation_roll (γ_d), all in units of degrees, and
- sphere coordinates (ϕ_d, θ_d) relative to the local coordinate axes.

Outputs of this process are:

- sphere coordinates (ϕ', θ') relative to the global coordinate axes.

The outputs are derived as follows:

$$\begin{aligned}
 \phi &= \phi_d * \pi \div 180 \\
 \theta &= \theta_d * \pi \div 180 \\
 \alpha &= \alpha_d * \pi \div 180 \\
 \beta &= \beta_d * \pi \div 180 \\
 \gamma &= \gamma_d * \pi \div 180 \\
 x_1 &= \text{Cos}(\phi) * \text{Cos}(\theta) \\
 y_1 &= \text{Sin}(\phi) * \text{Cos}(\theta) \\
 z_1 &= \text{Sin}(\theta) \\
 x_2 &= \text{Cos}(\beta) * \text{Cos}(\alpha) * x_1 - \text{Cos}(\beta) * \text{Sin}(\alpha) * y_1 + \text{Sin}(\beta) * z_1 \\
 y_2 &= (\text{Cos}(\gamma) * \text{Sin}(\alpha) + \text{Sin}(\gamma) * \text{Sin}(\beta) * \text{Cos}(\alpha)) * x_1 + \\
 &\quad (\text{Cos}(\gamma) * \text{Cos}(\alpha) - \text{Sin}(\gamma) * \text{Sin}(\beta) * \text{Sin}(\alpha)) * y_1 - \\
 &\quad \text{Sin}(\gamma) * \text{Cos}(\beta) * z_1 \\
 z_2 &= (\text{Sin}(\gamma) * \text{Sin}(\alpha) - \text{Cos}(\gamma) * \text{Sin}(\beta) * \text{Cos}(\alpha)) * x_1 + \\
 &\quad (\text{Sin}(\gamma) * \text{Cos}(\alpha) + \text{Cos}(\gamma) * \text{Sin}(\beta) * \text{Sin}(\alpha)) * y_1 + \\
 &\quad \text{Cos}(\gamma) * \text{Cos}(\beta) * z_1 \\
 \phi' &= \text{Atan2}(y_2, x_2) * 180 \div \pi \\
 \theta' &= \text{Asin}(z_2) * 180 \div \pi
 \end{aligned}
 \tag{41}$$

8.15.1.4 Conversion of sample locations for rectangular region-wise packing

Inputs to this process are:

- sample location (x, y) within the packed region, where x and y are in relative region-wise packed picture sample units, while the sample location is at an integer sample location within the packed picture,
- the width and the height (projRegWidth, projRegHeight) of the projected region, in relative projected picture sample units,
- the width and the height (packedRegWidth, packedRegHeight) of the packed region, in relative region-wise packed picture sample units,
- transform type (transformType), and
- offset values for the sampling position (offsetX, offsetY) in the range of 0, inclusive, to 1, exclusive, in horizontal and vertical relative region-wise packed picture sample units, respectively.

NOTE offsetX and offsetY both being equal to 0.5 indicates a sampling position that is at the centre point of a sample in packed picture sample units.

Outputs of this process are:

- the centre point of the sample location (hPos, vPos) within the projected region in relative projected picture sample units, where hPos and vPos could have non-integer real values.

The outputs are derived as follows:

```

if( transformType == 0 || transformType == 1 || transformType == 2 || transform-
Type == 3 ) {
    horRatio = projRegWidth ÷ packedRegWidth
    verRatio = projRegHeight ÷ packedRegHeight
} else if ( transformType == 4 || transformType == 5 || transformType == 6 ||
transformType == 7 ) {
    horRatio = projRegWidth ÷ packedRegHeight
    verRatio = projRegHeight ÷ packedRegWidth
}
if( transformType == 0 ) {
    hPos = horRatio * ( x + offsetX )
    vPos = verRatio * ( y + offsetY )
} else if ( transformType == 1 ) {
    hPos = horRatio * ( packedRegWidth - x - offsetX )
    vPos = verRatio * ( y + offsetY )
} else if ( transformType == 2 ) {
    hPos = horRatio * ( packedRegWidth - x - offsetX )
    vPos = verRatio * ( packedRegHeight - y - offsetY )
} else if ( transformType == 3 ) {
    hPos = horRatio * ( x + offsetX )
    vPos = verRatio * ( packedRegHeight - y - offsetY )
} else if ( transformType == 4 ) {
    hPos = horRatio * ( y + offsetY )
    vPos = verRatio * ( x + offsetX )
} else if ( transformType == 5 ) {
    hPos = horRatio * ( y + offsetY )
    vPos = verRatio * ( packedRegWidth - x - offsetX )
} else if ( transformType == 6 ) {
    hPos = horRatio * ( packedRegHeight - y - offsetY )
    vPos = verRatio * ( packedRegWidth - x - offsetX )
} else if ( transformType == 7 ) {
    hPos = horRatio * ( packedRegHeight - y - offsetY )
    vPos = verRatio * ( x + offsetX )
}

```

(42)

8.15.1.5 Mapping of luma sample locations within a cropped decoded picture to sphere coordinates relative to the global coordinate axes

This subclause specifies the semantics of luma sample locations within a cropped decoded picture to sphere coordinates relative to the global coordinate axes.

offsetX is set equal to 0.5 and offsetY is set equal to 0.5.

If RegionWisePackingFlag is equal to 1, the following applies for each packed region n in the range of 0 to NumPackedRegions - 1, inclusive:

- For each sample location (xPackedPicture, yPackedPicture) belonging to the n -th packed region, the following applies:
 - The corresponding sample location (xProjPicture, yProjPicture) of the projected picture is derived as follows:
 - x is set equal to $xPackedPicture - PackedRegionLeft[n]$.
 - y is set equal to $yPackedPicture - PackedRegionTop[n]$.
 - [Subclause 8.15.1.4](#) is invoked with x , y , PackedRegionWidth[n], PackedRegionHeight[n], ProjRegionWidth[n], ProjRegionHeight[n], TransformType[n], offsetX and offsetY as inputs, and the output is assigned to sample location (hPos, vPos).

- $x_{ProjPicture}$ is set equal to $ProjRegionLeft[n] + hPos$.
- When $StereoFlag$ is equal to 0 or $TopBottomFlag$ is equal to 1, and when $x_{ProjPicture}$ is greater than or equal to $rwp_proj_picture_width$, $x_{ProjPicture}$ is set equal to $x_{ProjPicture} - rwp_proj_picture_width$.
- When $SideBySideFlag$ is equal to 1, the following applies:
 - When $ProjRegionLeft[n]$ is less than $rwp_proj_picture_width / 2$ and $x_{ProjPicture}$ is greater than or equal to $rwp_proj_picture_width / 2$, $x_{ProjPicture}$ is set equal to $x_{ProjPicture} - rwp_proj_picture_width / 2$.
 - When $ProjRegionLeft[n]$ is greater than or equal to $rwp_proj_picture_width / 2$ and $x_{ProjPicture}$ is greater than or equal to $rwp_proj_picture_width$, $x_{ProjPicture}$ is set equal to $x_{ProjPicture} - rwp_proj_picture_width / 2$.
- $y_{ProjPicture}$ is set equal to $ProjRegionTop[n] + vPos$.
- [Subclause 8.15.1.6](#) is invoked with $x_{ProjPicture}$, $y_{ProjPicture}$, $ConstituentPicWidth$, and $ConstituentPicHeight$ as inputs, and the outputs indicating the sphere coordinates and the constituent picture index (for frame-packed stereoscopic video) for the luma sample location ($x_{PackedPicture}$, $y_{PackedPicture}$) belonging to the n -th packed region in the decoded picture.

Otherwise if $RegionWisePackingFlag$ is equal 0 and $CmpFlag$ is equal to 1, the following applies for each sample location (x, y) that is not a cubemap projection guard band sample within the cropped decoded picture:

- $x_{ProjPicture}$ is set equal to $x + offsetX$.
- $y_{ProjPicture}$ is set equal to $y + offsetY$.
- [Subclause 8.15.1.6](#) is invoked with $x_{ProjPicture}$, $y_{ProjPicture}$, $ConstituentPicWidth$, and $ConstituentPicHeight$ as inputs, and the outputs indicating the sphere coordinates and the constituent picture index (for frame-packed stereoscopic video) for the sample location (x, y) within the cropped decoded picture.

Otherwise ($RegionWisePackingFlag$ is equal to 0, and $CmpFlag$ is equal to 0), the following applies for each sample location (x, y) that is not an equirectangular projection guard band sample within the cropped decoded picture, where a sample location (x, y) is an equirectangular projection guard band sample when and only when $ErpFlag$ is equal to 1, x is in the range of 0 to $erp_left_guard_band_width - 1$, inclusive, or $ConstituentPicWidth - erp_right_guard_band_width$ to $ConstituentPicWidth - 1$, inclusive, and y is in the range of 0 to $ConstituentPicHeight - 1$, inclusive:

- $x_{ProjPicture}$ is set equal to $x + offsetX$.
- $y_{ProjPicture}$ is set equal to $y + offsetY$.
- If $ErpFlag$ is equal to 0, $projPicWidth$ is set equal to $ConstituentPicWidth$. Otherwise ($ErpFlag$ is equal to 1), $projPicWidth$ is set equal to $ConstituentPicWidth - (erp_left_guard_band_width + erp_right_guard_band_width)$.
- [Subclause 8.15.1.6](#) is invoked with $x_{ProjPicture}$, $y_{ProjPicture}$, $projPicWidth$, and $ConstituentPicHeight$ as inputs, and the outputs indicating the sphere coordinates and the constituent picture index (for frame-packed stereoscopic video) for the sample location (x, y) within the region-wise packed picture.

8.15.1.6 Conversion from a sample location in a projected picture to sphere coordinates relative to the global coordinate axes

Inputs to this process are:

- the centre point of a sample location ($x_{\text{ProjPicture}}$, $y_{\text{ProjPicture}}$) within a projected picture, where $x_{\text{ProjPicture}}$ and $y_{\text{ProjPicture}}$ are in relative projected picture sample units and could have non-integer real values, and
- pictureWidth and pictureHeight , which are the width and height, respectively, of a monoscopic projected luma picture, in relative projected picture sample units.

Outputs of this process are:

- sphere coordinates (azimuthGlobal , elevationGlobal), in units of degrees relative to the global coordinate axes, and
- when StereoFlag is equal to 1, the index of the constituent picture ($\text{constituentPicture}$) equal to 0 or 1.

The outputs are derived with the following ordered steps:

1. $\text{constituentPicture}$, $x_{\text{ProjPicture}}$, and $y_{\text{ProjPicture}}$ are conditionally set as follows:
 - If $x_{\text{ProjPicture}}$ is greater than or equal to pictureWidth or $y_{\text{ProjPicture}}$ is greater than or equal to pictureHeight , the following applies:
 - $\text{constituentPicture}$ is set equal to 1.
 - When $x_{\text{ProjPicture}}$ is greater than or equal to pictureWidth , $x_{\text{ProjPicture}}$ is set to $x_{\text{ProjPicture}} - \text{pictureWidth}$.
 - When $y_{\text{ProjPicture}}$ is greater than or equal to pictureHeight , $y_{\text{ProjPicture}}$ is set to $y_{\text{ProjPicture}} - \text{pictureHeight}$.
 - Otherwise, $\text{constituentPicture}$ is set equal to 0.
2. [Subclause 8.15.1.2](#) is invoked with pictureWidth , pictureHeight , $x_{\text{ProjPicture}}$, and $y_{\text{ProjPicture}}$ as inputs, and the output is assigned to azimuthLocal , elevationLocal .
3. azimuthGlobal and elevationGlobal are set as follows:
 - If RotationFlag is equal to 1, [subclause 8.15.1.3](#) is invoked with azimuthLocal , elevationLocal , RotationYaw , RotationPitch , and RotationRoll as inputs, and the output is assigned to azimuthGlobal and elevationGlobal .
 - Otherwise, azimuthGlobal is set equal to azimuthLocal and elevationGlobal is set equal to elevationLocal .

8.15.1.7 Calculation of the cubemap face size for a projected picture

Inputs to this process are:

- pictureWidth and pictureHeight , which are the width and height, respectively, of a monoscopic projected luma picture, in relative projected picture sample units.

Outputs of this process are:

- faceWidth and faceHeight , which are the width and height, respectively, of a projected face, in relative projected picture sample units.

The outputs are derived as follows:

```

gcmpPicWidth = pictureWidth
gcmpPicHeight = pictureHeight
gcmpGuardBandSamples = gcmp_guard_band_flag ? gcmp_guard_band_samples_
minus1 + 1 : 0
if( gcmp_guard_band_flag && gcmp_guard_band_boundary_exterior_flag ) {
    gcmpPicWidth = pictureWidth - 2 * gcmpGuardBandSamples
    gcmpPicHeight = pictureHeight - 2 * gcmpGuardBandSamples
}
if( gcmp_packing_type == 0 ) {
    if( gcmp_guard_band_flag )
        gcmpPicHeight -= 2 * gcmpGuardBandSamples
    faceWidth = gcmpPicWidth
    faceHeight = gcmpPicHeight / 6
} else if( gcmp_packing_type == 1 ) {
    if( gcmp_guard_band_flag )
        gcmpPicWidth -= 2 * gcmpGuardBandSamples
    faceWidth = gcmpPicWidth / 2
    faceHeight = gcmpPicHeight / 3
} else if( gcmp_packing_type == 2 ) {
    if( gcmp_guard_band_flag )
        gcmpPicHeight -= 2 * gcmpGuardBandSamples
    faceWidth = gcmpPicWidth / 3
    faceHeight = gcmpPicHeight / 2
} else if( gcmp_packing_type == 3 ) {
    if( gcmp_guard_band_flag )
        gcmpPicWidth -= 2 * gcmpGuardBandSamples
    faceWidth = gcmpPicWidth / 6
    faceHeight = gcmpPicHeight
} else if( gcmp_packing_type == 4 ) {
    if( gcmp_guard_band_flag )
        gcmpPicWidth -= 2 * gcmpGuardBandSamples
    faceWidth = gcmpPicWidth / 3
    faceHeight = gcmpPicHeight
} else if( gcmp_packing_type == 5 ) {
    if( gcmp_guard_band_flag )
        gcmpPicHeight -= 2 * gcmpGuardBandSamples
    faceWidth = gcmpPicWidth
    faceHeight = gcmpPicHeight / 3
}

```

(43)

The values of `faceWidth` and `faceHeight` are constrained as follows:

- If `gcmp_packing_type` is equal to 4, the following constraints apply:
 - When `ChromaFormatIdc` is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), `faceWidth` shall be a multiple of 4 in units of luma samples.
 - When `ChromaFormatIdc` is equal to 1 (4:2:0 chroma format), `faceHeight` shall be a multiple of 2 in units of luma samples.
- Otherwise, if `gcmp_packing_type` is equal to 5, the following constraints apply:
 - When `ChromaFormatIdc` is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), `faceWidth` shall be a multiple of 2 in units of luma samples.

- When ChromaFormatIdc is equal to 1 (4:2:0 chroma format), faceHeight shall be a multiple of 4 in units of luma samples.
- Otherwise, the following constraints apply:
 - When ChromaFormatIdc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), faceWidth shall be a multiple of 2 in units of luma samples.
 - When ChromaFormatIdc is equal to 1 (4:2:0 chroma format), faceHeight shall be a multiple of 2 in units of luma samples.

It is a requirement of bitstream conformance that the following constraints apply:

- If `gcmp_packing_type` is equal to 0, `gcmpPicHeight` shall be a multiple of 6, and `gcmpPicWidth` shall be equal to `gcmpPicHeight / 6`.
- Otherwise, if `gcmp_packing_type` is equal to 1, `gcmpPicWidth` shall be a multiple of 2 and `gcmpPicHeight` shall be a multiple of 3, and `gcmpPicWidth / 2` shall be equal to `gcmpPicHeight / 3`.
- Otherwise, if `gcmp_packing_type` is equal to 2, `gcmpPicWidth` shall be a multiple of 3 and `gcmpPicHeight` shall be a multiple of 2, and `gcmpPicWidth / 3` shall be equal to `gcmpPicHeight / 2`.
- Otherwise, if `gcmp_packing_type` is equal to 3, `gcmpPicWidth` shall be a multiple of 6, and `gcmpPicWidth / 6` shall be equal to `gcmpPicHeight`.
- Otherwise, if `gcmp_packing_type` is equal to 4, `gcmpPicWidth` shall be a multiple of 6, and `gcmpPicWidth / 3` shall be equal to `gcmpPicHeight`.
- Otherwise, if `gcmp_packing_type` is equal to 5, `gcmpPicHeight` shall be a multiple of 6, and `gcmpPicWidth` shall be equal to `gcmpPicHeight / 3`.

8.15.1.8 Conversion from a sample location in a projected picture to a sample location in a projected cubemap face

Inputs to this process are:

- sample location (`hPos`, `vPos`) within the projected picture in relative projected picture sample units, where `hPos` and `vPos` could have non-integer real values, and
- `faceWidth` and `faceHeight`, which are the width and height, respectively, of the projected face, in relative projected picture sample units.

Outputs of this process are:

- the sample location (`hPosFace`, `vPosFace`) within the projected face in relative projected picture sample units, where `hPosFace` and `vPosFace` could have non-integer real values.

The outputs are derived as follows:

```

gbSize = gcmpGuardBandSamples
tmpHorPos = hPos
tmpVerPos = vPos
if( gcmp_guard_band_flag ) {
  if( gcmp_guard_band_boundary_exterior_flag ) {
    tmpHorPos = hPos - gbSize
    tmpVerPos = vPos - gbSize
  }
  if( gcmp_packing_type == 0 )
    tmpVerPos = tmpVerPos < 3 * faceHeight ? tmpVerPos : tmpVerPos - 2 * gbSize
  else if( gcmp_packing_type == 1 )
    tmpHorPos = tmpHorPos < faceWidth ? tmpHorPos : tmpHorPos - 2 * gbSize
  else if( gcmp_packing_type == 2 )
    tmpVerPos = tmpVerPos < faceHeight ? tmpVerPos : tmpVerPos - 2 * gbSize
  else if( gcmp_packing_type == 3 )
    tmpHorPos = tmpHorPos < 3 * faceWidth ? tmpHorPos : tmpHorPos - 2 * gbSize
  else if( gcmp_packing_type == 4 )
    tmpHorPos = tmpHorPos < faceWidth / 2 ? tmpHorPos : tmpHorPos < 2.5 * faceWidth +
gbSize ?
    tmpHorPos - gbSize : tmpHorPos - 2 * gbSize
  else if( gcmp_packing_type == 5 )
    tmpVerPos = tmpVerPos < faceHeight / 2 ? tmpVerPos : tmpVerPos < 2.5 * faceHeight +
gbSize ?
    tmpVerPos - gbSize : tmpVerPos - 2 * gbSize
}
w = Floor( tmpHorPos ÷ faceWidth )
h = Floor( tmpVerPos ÷ faceHeight )
hPosFace = tmpHorPos - w * faceWidth
vPosFace = tmpVerPos - h * faceHeight

```

(44)

8.15.1.9 Rotation of sample locations for a projected cubemap face

Inputs to this process are:

- sample location (hPosFace, vPosFace) within the n-th projected face in relative projected picture sample units, where hPosFace and vPosFace could have non-integer real values, and
- faceWidth and faceHeight, which are the width and height, respectively, of the projected face, in relative projected picture sample units.

Outputs of this process are:

- the rotated sample location (hPosRot, vPosRot) within the projected face in relative projected picture sample units, where hPosRot and vPosRot could have non-integer real values.

The outputs are derived as follows:

```

if( gcmp_face_rotation[ n ] == 0 ) {
    hPosRot = hPosFace
    vPosRot = vPosFace
} else if( gcmp_face_rotation[ n ] == 1 ) {
    hPosRot = vPosFace
    vPosRot = faceWidth - hPosFace
} else if( gcmp_face_rotation[ n ] == 2 ) {
    hPosRot = faceWidth - hPosFace
    vPosRot = faceHeight - vPosFace
} else if( gcmp_face_rotation[ n ] == 3 ) {
    hPosRot = faceHeight - vPosFace
    vPosRot = hPosFace
}

```

(45)

8.15.1.10 Adjustment of a sample location for hemisphere cubemap projection

Inputs to this process are:

- sample location (hPosRot, vPosRot) within the n-th projected face in relative projected picture sample units, where hPosRot and vPosRot could have non-integer real values, and
- faceWidth and faceHeight, which are the width and height, respectively, of the projected face, in relative projected picture sample units.

Outputs of this process are:

- the adjusted sample location (hPosAdj, vPosAdj) within the n-th projected face in relative projected picture sample units, where hPosAdj and vPosAdj could have non-integer real values.

The outputs are derived as follows:

```

leftFaceIdx = {5, 3, 1, 0, 0, 1}
rightFaceIdx = {4, 2, 0, 1, 1, 0}
topFaceIdx = {2, 4, 4, 4, 2, 2}
bottomFaceIdx = {3, 5, 5, 5, 3, 3}
hPosAdj = hPosRot
vPosAdj = vPosRot
if( n != 2 )
    if( face_index[2] == leftFaceIdx[ face_index[ n ] ] && hPosAdj >= faceWidth / 2 )
        hPosAdj -= faceWidth / 2
    else if( face_index[2] == rightFaceIdx[ face_index[ n ] ] && hPosAdj < faceWidth / 2 )
        hPosAdj += faceWidth / 2
    else if( face_index[2] == topFaceIdx[ face_index[ n ] ] && vPosAdj >= faceHeight / 2 )
        vPosAdj -= faceHeight / 2
    else if( face_index[2] == bottomFaceIdx[ face_index[ n ] ] && vPosAdj < faceHeight / 2 )
        vPosAdj += faceHeight / 2

```

(46)

8.15.2 Equirectangular projection SEI message

8.15.2.1 Equirectangular projection SEI message syntax

equirectangular_projection(payloadSize) {	Descriptor
erp_cancel_flag	u(1)
if(!erp_cancel_flag) {	
erp_persistence_flag	u(1)
erp_guard_band_flag	u(1)

erp_reserved_zero_2bits	u(2)
if(erp_guard_band_flag == 1) {	
erp_guard_band_type	u(3)
erp_left_guard_band_width	u(8)
erp_right_guard_band_width	u(8)
}	
}	
}	

8.15.2.2 Equirectangular projection SEI message semantics

The equirectangular projection SEI message provides information to enable remapping (through an equirectangular projection) of the colour samples of the projected pictures onto a sphere coordinate space in sphere coordinates (ϕ , θ) for use in omnidirectional video applications for which the viewing perspective is from the origin looking outward toward the inside of the sphere. The sphere coordinates are defined so that ϕ is the azimuth (longitude, increasing eastward) and θ is the elevation (latitude, increasing northward).

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

- A chroma format indicator, denoted herein by `ChromaFormatIdc`, as described in [subclause 7.3](#).

When an equirectangular projection SEI message is present for any picture of a CLVS, an equirectangular projection SEI message shall be present for the first picture of the CLVS and no SEI message indicating a different type of projection shall be present for any picture of the CLVS.

When the SAR for a picture is indicated by `vui_aspect_ratio_idc` or `sari_aspect_ratio_idc` greater than 1, there should be no equirectangular projection SEI messages applicable for the picture.

A frame packing arrangement SEI message for which all the following conditions are true is referred to as an effectively applicable frame packing arrangement SEI message:

- The value of `fp_arrangement_cancel_flag` is equal to 0.
- The value of `fp_arrangement_type` is equal to 3, 4, or 5.
- The value of `fp_quincunx_sampling_flag` is equal to 0.
- The value of `fp_spatial_flipping_flag` is equal to 0.
- The value of `fp_field_views_flag` is equal to 0.
- The value of `fp_frame0_grid_position_x` is equal to 0.
- The value of `fp_frame0_grid_position_y` is equal to 0.
- The value of `fp_frame1_grid_position_x` is equal to 0.
- The value of `fp_frame1_grid_position_y` is equal to 0.

When a frame packing arrangement SEI message with `fp_arrangement_cancel_flag` equal to 0 that applies to the picture is present that is not an effectively applicable frame packing arrangement SEI message, an equirectangular projection SEI message with `erp_cancel_flag` equal to 0 that applies to the picture shall not be present. Decoders shall ignore equirectangular projection SEI messages when a frame packing arrangement SEI message with `fp_arrangement_cancel_flag` equal to 0 that applies to the picture is present that is not an effectively applicable frame packing arrangement SEI message.

erp_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous equirectangular projection SEI message in output order. **erp_cancel_flag** equal to 0 indicates that equirectangular projection information follows.

erp_persistence_flag specifies the persistence of the equirectangular projection SEI message for the current layer.

erp_persistence_flag equal to 0 specifies that the equirectangular projection SEI message applies to the current decoded picture only.

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

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

erp_guard_band_flag equal to 1 indicates that the constituent picture contains guard band areas for which the sizes are specified by the syntax elements **erp_left_guard_band_width** and **erp_right_guard_band_width**. **erp_guard_band_flag** equal to 0 indicates that the constituent picture does not contain guard band areas for which the sizes are specified by the syntax elements **erp_left_guard_band_width** and **erp_right_guard_band_width**.

erp_reserved_zero_2bits shall be equal to 0 in bitstreams conforming to this edition of this document. Other values for **erp_reserved_zero_2bits** are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of **erp_reserved_zero_2bits**.

erp_guard_band_type indicates the type of the guard bands as follows:

- **erp_guard_band_type** equal to 0 indicates that the content of the guard band in relation to the content of the constituent picture is unknown or unspecified or specified by other means not specified in this document.
- **erp_guard_band_type** equal to 1 indicates that the content of the guard band suffices for interpolation of sample values at sub-pel sample fractional locations within the constituent picture.

NOTE **erp_guard_band_type** equal to 1 could be used when the source boundary samples of a constituent picture have been copied horizontally to the guard band.

- **erp_guard_band_type** equal to 2 indicates that the content of the guard band represents actual picture content at a quality that gradually changes from the picture quality of the constituent picture to that of the spherically adjacent region.
- **erp_guard_band_type** equal to 3 indicates that the content of the guard bands represents actual picture content at a similar level of quality as the constituent picture.
- **erp_guard_band_type** values greater than 3 are reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value of **erp_guard_band_type** when the value is greater than 3 as equivalent to the value 0.

erp_left_guard_band_width specifies the width of the guard band on the left side of the constituent picture in units of luma samples. When **erp_guard_band_flag** is equal to 0, the value of **erp_left_guard_band_width** is inferred to be equal to 0. When **ChromaFormatIdc** is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), **erp_left_guard_band_width** shall be an even number.

erp_right_guard_band_width specifies the width of the guard band on the right side of the constituent picture in units of luma samples. When **erp_guard_band_flag** is equal to 0, the value of **erp_right_guard_**

band_width is inferred to be equal to 0. When ChromaFormatIdc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), erp_right_guard_band_width shall be an even number.

8.15.3 Generalized cubemap projection SEI message

8.15.3.1 Generalized cubemap projection SEI message syntax

generalized_cubemap_projection(payloadSize) {	Descriptor
gcmp_cancel_flag	u(1)
if(!gcmp_cancel_flag) {	
gcmp_persistence_flag	u(1)
gcmp_packing_type	u(3)
gcmp_mapping_function_type	u(2)
for(i = 0; i < (gcmp_packing_type == 4 gcmp_packing_type == 5) ? 5 : 6; i++) {	
gcmp_face_index[i]	u(3)
gcmp_face_rotation[i]	u(2)
if(gcmp_mapping_function_type == 2) {	
gcmp_function_coeff_u[i]	u(7)
gcmp_function_u_affected_by_v_flag[i]	u(1)
gcmp_function_coeff_v[i]	u(7)
gcmp_function_v_affected_by_u_flag[i]	u(1)
}	
}	
gcmp_guard_band_flag	u(1)
if(gcmp_guard_band_flag) {	
gcmp_guard_band_type	u(3)
gcmp_guard_band_boundary_exterior_flag	u(1)
gcmp_guard_band_samples_minus1	u(4)
}	
}	
}	

8.15.3.2 Generalized cubemap projection SEI message semantics

The generalized cubemap projection SEI message provides information to enable remapping (through a generalized cubemap projection) of the colour samples of the projected pictures onto a sphere coordinate space in sphere coordinates (ϕ , θ) for use in omnidirectional video applications for which the viewing perspective is from the origin looking outward toward the inside of the sphere. The sphere coordinates are defined so that ϕ is the azimuth (longitude, increasing eastward) and θ is the elevation (latitude, increasing northward).

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

- A chroma format indicator, denoted herein by ChromaFormatIdc, as described in [subclause 7.3](#).

When a generalized cubemap projection SEI message is present for any picture of a CLVS, a generalized cubemap projection SEI message shall be present for the first picture of the CLVS and no SEI message indicating a different type of projection shall be present for any picture of the CLVS.

When the SAR for a picture is indicated by `vui_aspect_ratio_idc` or `sari_aspect_ratio_idc` greater than 1, there should be no generalized cubemap projection SEI messages applicable for the picture.

A frame packing arrangement SEI message for which all the following conditions are true is referred to as an effectively applicable frame packing arrangement SEI message:

- The value of `fp_arrangement_cancel_flag` is equal to 0.
- The value of `fp_arrangement_type` is equal to 3, 4, or 5.
- The value of `fp_quincunx_sampling_flag` is equal to 0.
- The value of `fp_spatial_flipping_flag` is equal to 0.
- The value of `fp_field_views_flag` is equal to 0.
- The value of `fp_frame0_grid_position_x` is equal to 0.
- The value of `fp_frame0_grid_position_y` is equal to 0.
- The value of `fp_frame1_grid_position_x` is equal to 0.
- The value of `fp_frame1_grid_position_y` is equal to 0.

When a frame packing arrangement SEI message with `fp_arrangement_cancel_flag` equal to 0 that applies to the picture is present that is not an effectively applicable frame packing arrangement SEI message, a generalized cubemap projection SEI message with `gcmp_cancel_flag` equal to 0 that applies to the picture shall not be present. Decoders shall ignore generalized cubemap projection SEI messages when a frame packing arrangement SEI message with `fp_arrangement_cancel_flag` equal to 0 that applies to the picture is present that is not an effectively applicable frame packing arrangement SEI message.

When all of the following conditions are true, the functionality of the generalized cubemap projection SEI message is exactly the same as the cubemap projection SEI message specified in Rec. ITU-T H.265 | ISO/IEC 23008-2 and Rec. ITU-T H.264 | ISO/IEC 14496-10:

- The value of `gcmp_packing_type` is equal to 2;
- The value of `gcmp_mapping_function_type` is equal to 0;
- The values of `gcmp_face_index[i]` for `i` from 0 to 5, inclusive, are equal to 5, 0, 4, 3, 1 and 2, respectively;
- The value of `gcmp_face_rotation[i]` is equal to 0 for each value of `i` in the range of 0 to 5, inclusive;
- The value of `gcmp_guard_band_flag` is equal to 0.

gcmp_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous generalized cubemap projection SEI message in output order. `gcmp_cancel_flag` equal to 0 indicates that cubemap projection information follows.

gcmp_persistence_flag specifies the persistence of the generalized cubemap projection SEI message for the current layer.

`gcmp_persistence_flag` equal to 0 specifies that the generalized cubemap projection SEI message applies to the current decoded picture only.

`gcmp_persistence_flag` equal to 1 specifies that the generalized cubemap projection SEI message applies to the current decoded picture and persists 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 cubemap projection SEI message is output that follows the current picture in output order.

gcmp_packing_type specifies the packing type and the position index of the cubemap packing as specified in [Table 10](#). When the value of **gcmp_packing_type** is in the range of 0 to 3, inclusive, cubemap packing with six faces is used. When **gcmp_packing_type** is 4 or 5, hemisphere cubemap packing with one full face and four half faces is used. The value of **gcmp_packing_type** shall be in the range of 0 to 5, inclusive. Other values for **gcmp_packing_type** are reserved for future use by ITU-T | ISO/IEC.

Table 10 — Specification of packing type and position index based on **gcmp_packing_type**

gcmp_packing_type	Packing type and position index						
0	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table>	0	1	2	3	4	5
0							
1							
2							
3							
4							
5							
1	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>1</td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td></tr> </table>	0	1	2	3	4	5
0	1						
2	3						
4	5						
2	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td><td>5</td></tr> </table>	0	1	2	3	4	5
0	1	2					
3	4	5					
3	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	0	1	2	3	4	5
0	1	2	3	4	5		
4	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> </table>	0	1	2	3	4	
0	1	2	3	4			
5	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> </table>	0	1	2	3	4	
0							
1							
2							
3							
4							

gcmp_mapping_function_type specifies the mapping function used to adjust the sample locations of the cubemap projection. **gcmp_mapping_function_type** equal to 0 specifies that the same mapping function as specified for the cubemap projection SEI message in Rec. ITU-T H.265 | ISO/IEC 23008-2 and Rec. ITU-T H.264 | ISO/IEC 14496-10 is used. **gcmp_mapping_function_type** equal to 1 specifies that the equi-angular mapping function is applied to adjust the sample locations of the projected face, as defined in [subclause 8.15.1.2](#). **gcmp_mapping_function_type** equal to 2 specifies that the coefficients of the mapping function applied to adjust the sample locations of the *i*-th projected face are specified by the syntax elements **gcmp_function_coeff_u[i]**, **gcmp_function_u_affected_by_v_flag[i]**, **gcmp_function_coeff_v[i]**, and **gcmp_function_v_affected_by_u_flag[i]**. The value of **gcmp_mapping_function_type** shall be in the range of 0 to 2, inclusive.

gcmp_face_index[i] specifies the face index for position index *i* in **gcmp_packing_type** and the relationship between the global coordinates 3D (X, Y, Z) and the local coordinate 2D (u, v) as specified in [subclause 8.15.1.2](#).

When `gcmp_packing_type` is equal to 4 or 5, it is a requirement of bitstream conformance that the following constraints apply:

- If `gcmp_face_index[2]` is equal to 0 or 1, the value of `gcmp_face_index[i]` for `i` equal to 0, 1, 3 or 4 shall be in the range of 2 to 5, inclusive.
- Otherwise, if `gcmp_face_index[2]` is equal to 2 or 3, the value of `gcmp_face_index[i]` for `i` equal to 0, 1, 3 or 4 shall be 0, 1, 4, or 5.
- Otherwise, the value of `gcmp_face_index[i]` for `i` equal to 0, 1, 3 or 4 shall be in the range of 0 to 3, inclusive.

`gcmp_face_rotation[i]` specifies the rotation to be applied to the face on position index `i` as specified in [Table 11](#).

Table 11 — Specification of counterclockwise rotation angle based on `gcmp_face_rotation[i]`

<code>gcmp_face_rotation[i]</code>	Rotation angle in degree (anticlockwise)
0	0
1	90
2	180
3	270

When `gcmp_packing_type` is equal to 4, it is a requirement of bitstream conformance that the following constraints apply:

- If `gcmp_face_index[2]` is equal to 0 or 1, the value of `gcmp_face_rotation[i]` for `i` equal to 0, 1, 3 or 4 shall be 0 or 2.
- Otherwise, if `gcmp_face_index[2]` is equal to 2 or 3, when `gcmp_face_index[i]` is equal to 1, the value of `gcmp_face_rotation[i]` shall be 0 or 2, and when `gcmp_face_index[i]` is equal to 0, 4 or 5, the value of `gcmp_face_rotation[i]` shall be 1 or 3.
- Otherwise, when `gcmp_face_index[i]` is equal to 0, the value of `gcmp_face_rotation[i]` shall be 0 or 2, and when `gcmp_face_index[i]` is equal to 1, 2 or 3, the value of `gcmp_face_rotation[i]` shall be 1 or 3.

When `gcmp_packing_type` is equal to 5, it is a requirement of bitstream conformance that the following constraints apply:

- If `gcmp_face_index[2]` is equal to 0 or 1, the value of `gcmp_face_rotation[i]` for `i` equal to 0, 1, 3 or 4 shall be 1 or 3.
- Otherwise, if `gcmp_face_index[2]` is equal to 2 or 3, when `gcmp_face_index[i]` is equal to 1, the value of `gcmp_face_rotation[i]` shall be 1 or 3, and when `gcmp_face_index[i]` is equal to 0, 4 or 5, the value of `gcmp_face_rotation[i]` shall be 0 or 2.
- Otherwise, when `gcmp_face_index[i]` is equal to 0, the value of `gcmp_face_rotation[i]` shall be 1 or 3, and when `gcmp_face_index[i]` is equal to 1, 2 or 3, the value of `gcmp_face_rotation[i]` shall be 0 or 2.

`gcmp_function_coeff_u[i]` specifies the coefficient used in the cubemap mapping function of the `u`-axis of the `i`-th face. When `gcmp_function_coeff_u[i]` is not present, it is inferred to be equal to 0.

`gcmp_function_u_affected_by_v_flag[i]` equal to 1 indicates that the cubemap mapping function of the `u`-axis refers to the `v` position of the sample location. `gcmp_function_u_affected_by_v_flag[i]` equal to 0 indicates that the cubemap mapping function in `u`-axis does not refer to the `v` position of the sample location.

gcmp_function_coeff_v[i] specifies the coefficient used in the cubemap mapping function of the v-axis of the i-th face. When **gcmp_function_coeff_v[i]** is not present, it is inferred to be equal to 0.

gcmp_function_v_affected_by_u_flag[i] equal to 1 indicates that the cubemap mapping function of the v-axis refers to the u position of the sample location. **gcmp_function_v_affected_by_u_flag[i]** equal to 0 indicates that the cubemap mapping function in v-axis does not refer to the u position of the sample location.

gcmp_guard_band_flag equal to 0 indicates that the coded picture does not contain guard band areas. **gcmp_guard_band_flag** equal to 1 indicates that the coded picture contains guard band areas for which the sizes are specified by the syntax element **gcmp_guard_band_samples_minus1**.

gcmp_guard_band_type indicates the type of the guard bands as follows:

- **gcmp_guard_band_type** equal to 0 indicates that the content of the guard bands in relation to the content of the coded face is unknown or unspecified or specified by other means not specified in this document.
- **gcmp_guard_band_type** equal to 1 indicates that the content of the guard bands suffices for interpolation of sample values at sub-pel sample fractional locations within the coded face.

NOTE **gcmp_guard_band_type** equal to 1 could be used when the source boundary samples of a coded face have been copied horizontally or vertically to the guard band.

- **gcmp_guard_band_type** equal to 2 indicates that the content of the guard bands represents actual picture content that is spherically adjacent to the content in the coded face at quality that gradually changes from the picture quality of the coded face to that of the spherically adjacent region.
- **gcmp_guard_band_type** equal to 3 indicates that the content of the guard bands represents actual picture content that is spherically adjacent to the content in the coded face at a similar picture quality as within the coded face.
- **gcmp_guard_band_type** values greater than 3 are reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value of **gcmp_guard_band_type** when the value is greater than 3 as equivalent to the value 0.

gcmp_guard_band_boundary_exterior_flag indicates which face boundaries contain guard bands, as specified in [Table 12](#).

Table 12 — Specification of guard band boundary location based on **gcmp_packing_type and **gcmp_guard_band_boundary_exterior_flag****

gcmp_packing_type	gcmp_guard_band_boundary_exterior_flag	Location of guard band						
0	0	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="text-align: center;">0</td></tr> <tr><td style="text-align: center;">1</td></tr> <tr><td style="text-align: center;">2</td></tr> <tr style="background-color: #cccccc;"><td style="text-align: center;">3</td></tr> <tr><td style="text-align: center;">4</td></tr> <tr><td style="text-align: center;">5</td></tr> </table>	0	1	2	3	4	5
0								
1								
2								
3								
4								
5								

Table 12 (continued)

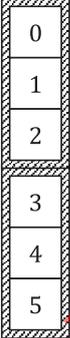
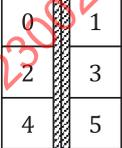
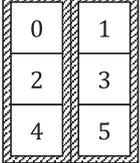
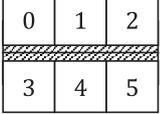
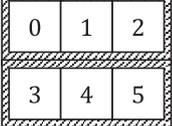
gcmp_packing_type	gcmp_guard_band_boundary_exterior_flag	Location of guard band
0	1	
1	0	
1	1	
2	0	
2	1	
3	0	
3	1	
4	0	
4	1	
5	0	

Table 12 (continued)

gcmp_packing_type	gcmp_guard_band_boundary_exterior_flag	Location of guard band
5	1	

gcmp_guard_band_samples_minus1 plus 1 specifies the number of guard band samples, in units of luma samples, used in the cubemap projected picture. When ChromaFormatIdc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), **gcmp_guard_band_samples_minus1** plus 1 shall correspond to an even number of luma samples within the cropped decoded picture.

8.15.4 Sphere rotation SEI message

8.15.4.1 Sphere rotation SEI message syntax

sphere_rotation(payloadSize) {	Descriptor
sphere_rotation_cancel_flag	u(1)
if(!sphere_rotation_cancel_flag) {	
sphere_rotation_persistence_flag	u(1)
sphere_rotation_reserved_zero_6bits	u(6)
yaw_rotation	i(32)
pitch_rotation	i(32)
roll_rotation	i(32)
}	
}	

8.15.4.2 Sphere rotation SEI message semantics

The sphere rotation SEI message provides information on rotation angles yaw (α), pitch (β), and roll (γ) that are used for conversion between the global coordinate axes and the local coordinate axes.

Relative to an (x, y, z) Cartesian coordinate system, yaw expresses a rotation around the z (vertical, up) axis, pitch rotates around the y (lateral, side-to-side) axis, and roll rotates around the x (back-to-front) axis. Rotations are extrinsic, i.e., around x, y, and z fixed reference axes. The angles increase clockwise when looking from the origin towards the positive end of an axis.

sphere_rotation_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous sphere rotation SEI message in output order. **sphere_rotation_cancel_flag** equal to 0 indicates that sphere rotation information follows.

sphere_rotation_persistence_flag specifies the persistence of the sphere rotation SEI message for the current layer.

sphere_rotation_persistence_flag equal to 0 specifies that the sphere rotation SEI message applies to the current decoded picture only.

sphere_rotation_persistence_flag equal to 1 specifies that the sphere rotation 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 sphere rotation SEI message is output that follows the current picture in output order.

When no omnidirectional video projection is indicated to apply to a picture, e.g., by an equirectangular projection SEI message with `erp_cancel_flag` equal to 0 or a generalized cubemap projection SEI message with `gcmp_cancel_flag` equal to 0 being present in the CLVS that applies to the picture, a sphere rotation SEI message with `sphere_rotation_cancel_flag` equal to 0 shall not be present in the CLVS that applies to the current picture. Decoders shall ignore sphere rotation SEI messages with `sphere_rotation_cancel_flag` equal to 0 for pictures to which no omnidirectional video projection is indicated to apply.

sphere_rotation_reserved_zero_6bits shall be equal to 0 in bitstreams conforming to this edition of this document. Other values for `sphere_rotation_reserved_zero_6bits` are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of `sphere_rotation_reserved_zero_6bits`.

yaw_rotation specifies the value of the yaw rotation angle, in units of 2^{-16} degrees. The value of `yaw_rotation` shall be in the range of $-180 * 2^{16}$ (i.e., -11 796 480) to $180 * 2^{16} - 1$ (i.e., 11 796 479), inclusive. When not present, the value of `yaw_rotation` is inferred to be equal to 0.

pitch_rotation specifies the value of the pitch rotation angle, in units of 2^{-16} degrees. The value of `pitch_rotation` shall be in the range of $-90 * 2^{16}$ (i.e., -5 898 240) to $90 * 2^{16}$ (i.e., 5 898 240), inclusive. When not present, the value of `pitch_rotation` is inferred to be equal to 0.

roll_rotation specifies the value of the roll rotation angle, in units of 2^{-16} degrees. The value of `roll_rotation` shall be in the range of $-180 * 2^{16}$ (i.e., -11 796 480) to $180 * 2^{16} - 1$ (i.e., 11 796 479), inclusive. When not present, the value of `roll_rotation` is inferred to be equal to 0.

8.15.5 Region-wise packing SEI message

8.15.5.1 Region-wise packing SEI message syntax

regionwise_packing(payloadSize) {	Descriptor
rwp_cancel_flag	u(1)
if(!rwp_cancel_flag) {	
rwp_persistence_flag	u(1)
rwp_constituent_picture_matching_flag	u(1)
rwp_reserved_zero_5bits	u(5)
rwp_num_packed_regions	u(8)
rwp_proj_picture_width	u(32)
rwp_proj_picture_height	u(32)
rwp_packed_picture_width	u(16)
rwp_packed_picture_height	u(16)
for(i = 0; i < rwp_num_packed_regions; i++) {	
rwp_reserved_zero_4bits[i]	u(4)
rwp_transform_type[i]	u(3)
rwp_guard_band_flag[i]	u(1)
rwp_proj_region_width[i]	u(32)
rwp_proj_region_height[i]	u(32)
rwp_proj_region_top[i]	u(32)
rwp_proj_region_left[i]	u(32)
rwp_packed_region_width[i]	u(16)

rwp_packed_region_height [i]	u(16)
rwp_packed_region_top [i]	u(16)
rwp_packed_region_left [i]	u(16)
if(rwp_guard_band_flag[i]) {	
rwp_left_guard_band_width [i]	u(8)
rwp_right_guard_band_width [i]	u(8)
rwp_top_guard_band_height [i]	u(8)
rwp_bottom_guard_band_height [i]	u(8)
rwp_guard_band_not_used_for_pred_flag [i]	u(1)
for(j = 0; j < 4; j++)	
rwp_guard_band_type [i][j]	u(3)
rwp_guard_band_reserved_zero_3bits [i]	u(3)
}	
}	
}	
}	

8.15.5.2 Region-wise packing SEI message semantics

The RWP SEI message provides information to enable remapping of the colour samples of the cropped decoded pictures onto projected pictures as well as information on the location and size of the guard bands, if any.

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

- A chroma format indicator, denoted herein by ChromaFormatIdc, as described in [subclause 7.3](#).

rwp_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous RWP SEI message in output order. **rwp_cancel_flag** equal to 0 indicates that RWP information follows.

rwp_persistence_flag specifies the persistence of the RWP SEI message for the current layer.

rwp_persistence_flag equal to 0 specifies that the RWP SEI message applies to the current decoded picture only.

rwp_persistence_flag equal to 1 specifies that the RWP 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 RWP SEI message is output that follows the current picture in output order.

When no omnidirectional video projection is indicated to apply to a picture, e.g., by an equirectangular projection SEI message with **erp_cancel_flag** equal to 0 or a generalized cubemap projection SEI message with **gcmp_cancel_flag** equal to 0 being present in the CLVS that applies to the current picture, a RWP SEI message with **rwp_cancel_flag** equal to 0 shall not be present in the CLVS that applies to the picture. Decoders shall ignore RWP SEI messages with **rwp_cancel_flag** equal to 0 for pictures to which no omnidirectional video projection is indicated to apply.

When an equirectangular projection SEI message with **erp_cancel_flag** equal to 0 and **erp_guard_band_flag** equal to 1 is present in the CLVS that applies to the current picture, a RWP SEI message with **rwp_cancel_flag** equal to 0 shall not be present in the CLVS that applies to the current picture.

When a generalized cubemap projection SEI message with `gmcp_cancel_flag` equal to 0 is present in the CLVS that applies to the current picture and precedes the RWP SEI message in decoding order, a RWP SEI message with `rwp_cancel_flag` equal to 0 shall not be present in the CLVS that applies to the current picture unless all the following conditions are true for the generalized cubemap projection SEI message:

- The value of `gcmp_packing_type` is equal to 2;
- The values of `gcmp_face_index[i]` for `i` from 0 to 5, inclusive, are equal to 5, 0, 4, 3, 1 and 2, respectively;
- The value of `gcmp_face_rotation[i]` is equal to 0 for each value of `i` in the range of 0 to 5, inclusive;
- The value of `gcmp_guard_band_flag` is equal to 0.

For the frame packing arrangement scheme indicated by a frame packing arrangement SEI message that applies to the current picture, if a RWP SEI message with `rwp_cancel_flag` equal to 0 is present that applies to the current picture, the frame packing arrangement scheme applies to the projected picture, otherwise, the frame packing arrangement scheme applies to the cropped decoded picture.

If a frame packing arrangement SEI message with `fp_arrangement_cancel_flag` equal to 0, `fp_arrangement_type` equal to 3, 4, or 5, and `fp_quincunx_sampling_flag` equal to 0 is not present that applies to the current picture, the variables `StereoFlag`, `TopBottomFlag`, `SideBySideFlag`, and `TempInterleavingFlag` are all set equal to 0, the variables `HorDiv1` and `VerDiv1` are both set equal to 1. Otherwise the following applies:

- `StereoFlag` is set equal to 1.
- When the `fp_arrangement_type` is equal to 3, `SideBySideFlag` is set equal to 1, `TopBottomFlag` and `TempInterleavingFlag` are both set equal to 0, `HorDiv1` is set equal to 2 and `VerDiv1` is set equal to 1.
- When the `fp_arrangement_type` is equal to 4, `TopBottomFlag` is set equal to 1, `SideBySideFlag` and `TempInterleavingFlag` are both set equal to 0, `HorDiv1` is set equal to 1 and `VerDiv1` is set equal to 2.
- When the `fp_arrangement_type` is equal to 5, `TempInterleavingFlag` is set equal to 1, `TopBottomFlag` and `SideBySideFlag` are both set equal to 0, `HorDiv1` and `VerDiv1` are both set equal to 1.

`rwp_constituent_picture_matching_flag` equal to 1 specifies that the projected region information, packed region information, and guard band region information in this SEI message apply individually to each constituent picture and that the packed picture and the projected picture have the same stereoscopic frame packing format indicated by the frame packing arrangement SEI message. `rwp_constituent_picture_matching_flag` equal to 0 specifies that the projected region information, packed region information, and guard band region information in this SEI message apply to the projected picture.

When either of the following conditions is true, the value of `rwp_constituent_picture_matching_flag` shall be equal to 0:

- `StereoFlag` is equal to 0.
- `StereoFlag` is equal to 1 and `fp_arrangement_type` is equal to 5.

`rwp_reserved_zero_5bits` shall be equal to 0 in bitstreams conforming to this edition of this document. Other values for `rwp_reserved_zero_5bits[i]` are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of `rwp_reserved_zero_5bits[i]`.

`rwp_num_packed_regions` specifies the number of packed regions when `rwp_constituent_picture_matching_flag` is equal to 0. The value of `rwp_num_packed_regions` shall be greater than 0. When `rwp_constituent_picture_matching_flag` is equal to 1, the total number of packed regions is equal to `rwp_num_packed_regions * 2`, and the information in each entry of the loop of `rwp_num_packed_regions` entries applies to each constituent picture of the projected picture and the packed picture.

rwp_proj_picture_width and **rwp_proj_picture_height** specify the width and height, respectively, of the projected picture, in relative projected picture sample units.

NOTE 1 Relative project picture sample unit is the unit used for the width or height of a projected picture or projected region. When a projected picture is a region-wise packed picture (i.e., there is a one-to-one mapping between the projected picture samples and the region-wise packed picture samples and a relative project picture sample unit is equivalent to a relative region-wise packed picture sample unit), **rwp_proj_picture_width** and **rwp_proj_picture_height** would have such values that **rwp_proj_picture_width** is an integer multiple of **cropPicWidth** and **rwp_proj_picture_height** is an integer multiple of **cropPicHeight**, where **cropPicWidth** and **cropPicHeight** are the width and height, respectively, of the cropped decoded picture, in units of luma samples.

The values of **rwp_proj_picture_width** and **rwp_proj_picture_height** shall both be greater than 0.

rwp_packed_picture_width and **rwp_packed_picture_height** specify the width and height, respectively, of the packed picture, in relative region-wise packed picture sample units.

The values of **rwp_packed_picture_width** and **rwp_packed_picture_height** shall both be greater than 0.

It is a requirement of bitstream conformance that **rwp_packed_picture_width** and **rwp_packed_picture_height** shall have such values that **rwp_packed_picture_width** is an integer multiple of **cropPicWidth** and **rwp_packed_picture_height** is an integer multiple of **cropPicHeight**, where **cropPicWidth** and **cropPicHeight** are the width and height, respectively, of the cropped decoded picture, in units of luma samples.

rwp_reserved_zero_4bits[i] shall be equal to 0 in bitstreams conforming to this edition of this document. Other values for **rwp_reserved_zero_4bits[i]** are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of **rwp_reserved_zero_4bits[i]**.

rwp_transform_type[i] specifies the rotation and mirroring to be applied to the *i*-th packed region to remap to the *i*-th projected region. When **rwp_transform_type[i]** specifies both rotation and mirroring, rotation applies before mirroring. The values of **rwp_transform_type[i]** are specified in [Table 13](#).

Table 13 — rwp_transform_type[i] values

Value	Description
0	no transform
1	mirroring horizontally
2	rotation by 180 degrees (anticlockwise)
3	rotation by 180 degrees (anticlockwise) before mirroring horizontally
4	rotation by 90 degrees (anticlockwise) before mirroring horizontally
5	rotation by 90 degrees (anticlockwise)
6	rotation by 270 degrees (anticlockwise) before mirroring horizontally
7	rotation by 270 degrees (anticlockwise)

rwp_guard_band_flag[i] equal to 0 specifies that the *i*-th packed region does not have a guard band. **rwp_guard_band_flag[i]** equal to 1 specifies that the *i*-th packed region has a guard band.

rwp_proj_region_width[i], **rwp_proj_region_height[i]**, **rwp_proj_region_top[i]** and **rwp_proj_region_left[i]** specify the width, height, top sample row, and the left-most sample column, respectively, of the *i*-th projected region, either within the projected picture (when **rwp_constituent_picture_matching_flag** is equal to 0) or within the constituent picture of the projected picture (when **rwp_constituent_picture_matching_flag** is equal to 1).

$\text{rwp_proj_region_width}[i]$, $\text{rwp_proj_region_height}[i]$, $\text{rwp_proj_region_top}[i]$, and $\text{rwp_proj_region_left}[i]$ are indicated in relative projected picture sample units.

NOTE 2 Two projected regions could partially or entirely overlap with each other.

$\text{rwp_packed_region_width}[i]$, $\text{rwp_packed_region_height}[i]$, $\text{rwp_packed_region_top}[i]$, and $\text{rwp_packed_region_left}[i]$ specify the width, height, the top luma sample row, and the left-most luma sample column, respectively, of the packed region, either within the region-wise packed picture (when $\text{rwp_constituent_picture_matching_flag}$ is equal to 0) or within each constituent picture of the region-wise packed picture (when $\text{rwp_constituent_picture_matching_flag}$ is equal to 1).

$\text{rwp_packed_region_width}[i]$, $\text{rwp_packed_region_height}[i]$, $\text{rwp_packed_region_top}[i]$, and $\text{rwp_packed_region_left}[i]$ are indicated in relative region-wise packed picture sample units. $\text{rwp_packed_region_width}[i]$, $\text{rwp_packed_region_height}[i]$, $\text{rwp_packed_region_top}[i]$, and $\text{rwp_packed_region_left}[i]$ shall represent integer horizontal and vertical coordinates of luma sample units within the cropped decoded pictures.

NOTE 3 Two packed regions could partially or entirely overlap with each other.

$\text{rwp_left_guard_band_width}[i]$ specifies the width of the guard band on the left side of the i -th packed region in relative region-wise packed picture sample units. When ChromaFormatIdc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), $\text{rwp_left_guard_band_width}[i]$ shall correspond to an even number of luma samples within the cropped decoded picture.

$\text{rwp_right_guard_band_width}[i]$ specifies the width of the guard band on the right side of the i -th packed region in relative region-wise packed picture sample units. When ChromaFormatIdc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), $\text{rwp_right_guard_band_width}[i]$ shall correspond to an even number of luma samples within the cropped decoded picture.

$\text{rwp_top_guard_band_height}[i]$ specifies the height of the guard band above the i -th packed region in relative region-wise packed picture sample units. When ChromaFormatIdc is equal to 1 (4:2:0 chroma format), $\text{rwp_top_guard_band_height}[i]$ shall correspond to an even number of luma samples within the cropped decoded picture.

$\text{rwp_bottom_guard_band_height}[i]$ specifies the height of the guard band below the i -th packed region in relative region-wise packed picture sample units. When ChromaFormatIdc is equal to 1 (4:2:0 chroma format), $\text{rwp_bottom_guard_band_height}[i]$ shall correspond to an even number of luma samples within the cropped decoded picture.

When $\text{rwp_guard_band_flag}[i]$ is equal to 1, $\text{rwp_left_guard_band_width}[i]$, $\text{rwp_right_guard_band_width}[i]$, $\text{rwp_top_guard_band_height}[i]$, or $\text{rwp_bottom_guard_band_height}[i]$ shall be greater than 0.

The i -th packed region as specified by this SEI message shall not overlap with any other packed region specified by the same SEI message or any guard band specified by the same SEI message.

The guard bands associated with the i -th packed region, if any, as specified by this SEI message shall not overlap with any packed region specified by the same SEI message or any other guard bands specified by the same SEI message.

$\text{rwp_guard_band_not_used_for_pred_flag}[i]$ equal to 0 specifies that the guard bands might or might not be used in the inter prediction process. $\text{rwp_guard_band_not_used_for_pred_flag}[i]$ equal to 1 specifies that the sample values of the guard bands are not used in the inter prediction process.

NOTE 4 When $\text{rwp_guard_band_not_used_for_pred_flag}[i]$ is equal to 1, the sample values within guard bands in cropped decoded pictures could be rewritten even if the cropped decoded pictures were used as references for inter prediction of subsequent pictures to be decoded. For example, the content of a packed region could be seamlessly expanded to its guard band with decoded and re-projected samples of another packed region.

rwp_guard_band_type[i][j] indicates the type of the guard bands for the i-th packed region as follows, with j equal to 0, 1, 2, or 3 indicating that the semantics apply to the left, right, top, or bottom edge, respectively, of the packed region:

- **rwp_guard_band_type**[i][j] equal to 0 indicates that the content of the guard bands in relation to the content of the packed regions is unknown or unspecified or specified by other means not specified in this document. When **rwp_guard_band_not_used_for_pred_flag**[i] is equal to 0, **rwp_guard_band_type**[i][j] shall not be equal to 0.
- **rwp_guard_band_type**[i][j] equal to 1 indicates that the content of the guard bands suffices for interpolation of sample values at sub-pel sample fractional locations within the packed region and less than one sample outside of the boundary of the packed region.

NOTE 5 **rwp_guard_band_type**[i][j] equal to 1 could be used when the boundary samples of a packed region have been copied horizontally or vertically to the guard band.

- **rwp_guard_band_type**[i][j] equal to 2 indicates that the content of the guard bands represents actual picture content that is spherically adjacent to the content in the packed region and is on the surface of the packed region at quality that gradually changes from the picture quality of the packed region to that of the spherically adjacent packed region.
- **rwp_guard_band_type**[i][j] equal to 3 indicates that the content of the guard bands represents actual picture content that is spherically adjacent to the content in the packed region and is on the surface of the packed region at a similar picture quality as within the packed region.
- **rwp_guard_band_type**[i][j] values greater than 3 are reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value of **rwp_guard_band_type**[i][j] when the value is greater than 3 as equivalent to the value 0.

rwp_guard_band_reserved_zero_3bits[i] shall be equal to 0 in bitstreams conforming to this edition of this document. Other values for **rwp_guard_band_reserved_zero_3bits**[i] are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of **rwp_guard_band_reserved_zero_3bits**[i].

The variables **NumPackedRegions**, **PackedRegionLeft**[n], **PackedRegionTop**[n], **PackedRegionWidth**[n], **PackedRegionHeight**[n], **ProjRegionLeft**[n], **ProjRegionTop**[n], **ProjRegionWidth**[n], **ProjRegionHeight**[n], and **TransformType**[n] are derived as follows:

- For n in the range of 0 to **rwp_num_packed_regions** – 1, inclusive, the following applies:
 - **PackedRegionLeft**[n] is set equal to **rwp_packed_region_left**[n].
 - **PackedRegionTop**[n] is set equal to **rwp_packed_region_top**[n].
 - **PackedRegionWidth**[n] is set equal to **rwp_packed_region_width**[n].
 - **PackedRegionHeight**[n] is set equal to **rwp_packed_region_height**[n].
 - **ProjRegionLeft**[n] is set equal to **rwp_proj_region_left**[n].
 - **ProjRegionTop**[n] is set equal to **rwp_proj_region_top**[n].
 - **ProjRegionWidth**[n] is set equal to **rwp_proj_region_width**[n].
 - **ProjRegionHeight**[n] is set equal to **rwp_proj_region_height**[n].

- TransformType[n] is set equal to rwp_transform_type[n].
- If rwp_constituent_picture_matching_flag is equal to 0, the following applies:
 - NumPackedRegions is set equal to rwp_num_packed_regions.
- Otherwise (rwp_constituent_picture_matching_flag is equal to 1), the following applies:
 - NumPackedRegions is set equal to 2 * rwp_num_packed_regions.
 - When TopBottomFlag is equal to 1, the following applies:
 - projLeftOffset and packedLeftOffset are both set equal to 0.
 - projTopOffset is set equal to rwp_proj_picture_height / 2 and packedTopOffset is set equal to rwp_packed_picture_height / 2.
 - When SideBySideFlag is equal to 1, the following applies:
 - projLeftOffset is set equal to rwp_proj_picture_width / 2 and packedLeftOffset is set equal to rwp_packed_picture_width / 2.
 - projTopOffset and packedTopOffset are both set equal to 0.
 - For n in the range of NumPackedRegions / 2 to NumPackedRegions - 1, inclusive, the following applies:
 - nIdx is set equal to n - NumPackedRegions / 2.
 - PackedRegionLeft[n] is set equal to rwp_packed_region_left[nIdx] + packedLeftOffset.
 - PackedRegionTop[n] is set equal to rwp_packed_region_top[nIdx] + packedTopOffset.
 - PackedRegionWidth[n] is set equal to rwp_packed_region_width[nIdx].
 - PackedRegionHeight[n] is set equal to rwp_packed_region_height[nIdx].
 - ProjRegionLeft[n] is set equal to rwp_proj_region_left[nIdx] + projLeftOffset.
 - ProjRegionTop[n] is set equal to rwp_proj_region_top[nIdx] + projTopOffset.
 - ProjRegionWidth[n] is set equal to rwp_proj_region_width[nIdx].
 - ProjRegionHeight[n] is set equal to rwp_proj_region_height[nIdx].
 - TransformType[n] is set equal to rwp_transform_type[nIdx].

For each value of n in the range of 0 to NumPackedRegions - 1, inclusive, the values of ProjRegionWidth[n], ProjRegionHeight[n], ProjRegionTop[n], and ProjRegionLeft[n] are constrained as follows:

- ProjRegionWidth[n] shall be in the range of 1 to rwp_proj_picture_width, inclusive.
- ProjRegionHeight[n] shall be in the range of 1 to rwp_proj_picture_height, inclusive.
- ProjRegionLeft[n] shall be in the range of 0 to rwp_proj_picture_width - 1, inclusive.
- ProjRegionTop[n] shall be in the range of 0 to rwp_proj_picture_height - 1, inclusive.
- If ProjRegionTop[n] is less than rwp_proj_picture_height / VerDiv1, the sum of ProjRegionTop[n] and ProjRegionHeight[n] shall be less than or equal to rwp_proj_picture_height / VerDiv1. Otherwise, the sum of ProjRegionTop[n] and ProjRegionHeight[n] shall be less than or equal to rwp_proj_picture_height / VerDiv1 * 2.

For each value of n in the range of 0 to $\text{NumPackedRegions} - 1$, inclusive, the values of $\text{PackedRegionWidth}[n]$, $\text{PackedRegionHeight}[n]$, $\text{PackedRegionTop}[n]$, and $\text{PackedRegionLeft}[n]$ are constrained as follows:

- $\text{PackedRegionWidth}[n]$ shall be in the range of 1 to $\text{rwp_packed_picture_width}$, inclusive.
- $\text{ProjRegionHeight}[n]$ shall be in the range of 1 to $\text{rwp_packed_picture_height}$, inclusive.
- $\text{PackedRegionLeft}[n]$ shall be in the range of 0 to $\text{rwp_packed_picture_width} - 1$, inclusive.
- $\text{PackedRegionTop}[n]$ shall be in the range of 0 to $\text{rwp_packed_picture_height} - 1$, inclusive.
- If $\text{PackedRegionLeft}[n]$ is less than $\text{rwp_packed_picture_width} / \text{HorDiv1}$, the sum of $\text{PackedRegionLeft}[n]$ and $\text{PackedRegionWidth}[n]$ shall be less than or equal to $\text{rwp_packed_picture_width} / \text{HorDiv1}$. Otherwise, the sum of $\text{PackedRegionLeft}[n]$ and $\text{PackedRegionWidth}[n]$ shall be less than or equal to $\text{rwp_packed_picture_width} / \text{HorDiv1} * 2$.
- If $\text{PackedRegionTop}[n]$ is less than $\text{rwp_packed_picture_height} / \text{VerDiv1}$, the sum of $\text{PackedRegionTop}[n]$ and $\text{PackedRegionHeight}[n]$ shall be less than or equal to $\text{rwp_packed_picture_height} / \text{VerDiv1}$. Otherwise, the sum of $\text{PackedRegionTop}[n]$ and $\text{PackedRegionHeight}[n]$ shall be less than or equal to $\text{rwp_packed_picture_height} / \text{VerDiv1} * 2$.
- When ChromaFormatIdc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format), $\text{PackedRegionLeft}[n]$ shall correspond to an even horizontal coordinate value of luma sample units, and $\text{PackedRegionWidth}[n]$ shall correspond to an even number of luma samples, both within the decoded picture.
- When ChromaFormatIdc is equal to 1 (4:2:0 chroma format), $\text{PackedRegionTop}[n]$ shall correspond to an even vertical coordinate value of luma sample units, and $\text{ProjRegionHeight}[n]$ shall correspond to an even number of luma samples, both within the decoded picture.

8.15.6 Omnidirectional viewport SEI message

8.15.6.1 Omnidirectional viewport SEI message syntax

omni_viewport(payloadSize) {	Descriptor
omni_viewport_id	u(10)
omni_viewport_cancel_flag	u(1)
if(!omni_viewport_cancel_flag) {	
omni_viewport_persistence_flag	u(1)
omni_viewport_cnt_minus1	u(4)
for(i = 0; i <= omni_viewport_cnt_minus1; i++) {	
omni_viewport_azimuth_centre[i]	i(32)
omni_viewport_elevation_centre[i]	i(32)
omni_viewport_tilt_centre[i]	i(32)
omni_viewport_hor_range[i]	u(32)
omni_viewport_ver_range[i]	u(32)
}	
}	
}	

8.15.6.2 Omnidirectional viewport SEI message semantics

The omnidirectional viewport SEI message specifies the coordinates of one or more regions of spherical-coordinate geometry, bounded by four great circles, corresponding to viewports recommended for

display when the user does not have control of the viewing orientation or has released control of the viewing orientation.

When an effectively applicable frame packing arrangement SEI message, as specified in [subclause 8.15.2.2](#) or [8.15.3.2](#), that applies to the picture is present, the information indicated by the omnidirectional viewport SEI message applies to both views.

omni_viewport_id contains an identifying number that may be used to identify the purpose of the one or more recommended viewport regions.

omni_viewport_id equal to 0 indicates that the recommended viewports are per "director's cut", i.e., a viewport suggested according to the creative intent of the content author or content provider. omni_viewport_id equal to 1 indicates that the recommended viewports are selected based on measurements of viewing statistics.

Values of omni_viewport_id from 2 to 511, inclusive, may be used as determined by the application. Values of omni_viewport_id from 512 to 1023 are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of omni_viewport_id in the range of 512 to 1023, inclusive, shall ignore it.

omni_viewport_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous omnidirectional viewport SEI message in output order. omni_viewport_cancel_flag equal to 0 indicates that omnidirectional viewport information follows.

omni_viewport_persistence_flag specifies the persistence of the omnidirectional viewport SEI message for the current layer.

omni_viewport_persistence_flag equal to 0 specifies that the omnidirectional viewport SEI message applies to the current decoded picture only.

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

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

When no omnidirectional video projection is indicated to apply to a picture, e.g., by an equirectangular projection SEI message with `erp_cancel_flag` equal to 0 or a generalized cubemap projection SEI message with `gcmp_cancel_flag` equal to 0 being present in the CLVS that applies to the current picture, an omnidirectional viewport SEI message with `omni_viewport_cancel_flag` equal to 0 shall not be present in the CLVS that applies to the picture. Decoders shall ignore omnidirectional viewport SEI messages with `omni_viewport_cancel_flag` equal to 0 for pictures to which no omnidirectional video projection is indicated to apply.

omni_viewport_cnt_minus1 plus 1 specifies the number of recommended viewport regions that are indicated by the SEI message.

When `omni_viewport_cnt_minus1` is greater than 0 and there is no information provided by external means not specified in this document on which recommended viewport is suggested to be displayed, the following applies:

- When `omni_viewport_id` is equal to 0 or 1, the 0-th recommended viewport is suggested to be displayed when the user does not have control of the viewing orientation or has released control of the viewing orientation.
- When `omni_viewport_id` is equal to 0, between any two recommended viewports per director's cut, the *i*-th recommended viewport has higher priority than the *j*-th recommended viewport for

any values of i and j when i is less than j . The 0-th recommended viewport per director's cut has the highest priority.

- When `omni_viewport_id` is equal to 1, between any two recommended viewports, the i -th recommended viewport has higher popularity, among some selection of candidate viewports, than the j -th recommended viewport for any values of i and j when i is less than j . The 0-th most-viewed recommended viewport has the highest popularity. The selection of the candidate viewports is outside the scope of this document.

`omni_viewport_azimuth_centre[i]` and `omni_viewport_elevation_centre[i]` indicate the centre of the i -th recommended viewport region, in units of 2^{-16} degrees relative to the global coordinate axes. The value of `omni_viewport_azimuth_centre[i]` shall be in the range of $-180 * 2^{16}$ (i.e., -11 796 480) to $180 * 2^{16} - 1$ (i.e., 11 796 479), inclusive. The value of `omni_viewport_elevation_centre[i]` shall be in the range of $-90 * 2^{16}$ (i.e., -5 898 240) to $90 * 2^{16}$ (i.e., 5 898 240), inclusive.

`omni_viewport_tilt_centre[i]` indicates the tilt angle of the i -th recommended viewport region, in units of 2^{-16} degrees. The value of `omni_viewport_tilt_centre[i]` shall be in the range of $-180 * 2^{16}$ (i.e., -11 796 480) to $180 * 2^{16} - 1$ (i.e., 11 796 479), inclusive.

`omni_viewport_hor_range[i]` indicates the azimuth range of the i -th recommended viewport region, in units of 2^{-16} degrees. The value of `omni_viewport_hor_range[i]` shall be in the range of 1 to $360 * 2^{16}$ (i.e., 23 592 960), inclusive.

`omni_viewport_ver_range[i]` indicates the elevation range of the i -th recommended viewport region, in units of 2^{-16} degrees. The value of `omni_viewport_ver_range[i]` shall be in the range of 1 to $180 * 2^{16}$ (i.e., 11 796 480), inclusive.

8.16 Frame-field information SEI message

8.16.1 Frame-field information SEI message syntax

frame_field_info(payloadSize) {	Descriptor
<code>ffi_field_pic_flag</code>	u(1)
if(<code>ffi_field_pic_flag</code>) {	
<code>ffi_bottom_field_flag</code>	u(1)
<code>ffi_pairing_indicated_flag</code>	u(1)
if(<code>ffi_pairing_indicated_flag</code>)	
<code>ffi_paired_with_next_field_flag</code>	u(1)
} else {	
<code>ffi_display_fields_from_frame_flag</code>	u(1)
if(<code>ffi_display_fields_from_frame_flag</code>)	
<code>ffi_top_field_first_flag</code>	u(1)
<code>ffi_display_elemental_periods_minus1</code>	u(8)
}	
<code>ffi_source_scan_type</code>	u(2)
<code>ffi_duplicate_flag</code>	u(1)
}	

8.16.2 Frame-field information SEI message semantics

The frame-field information SEI message may be used to indicate how the associated picture should be displayed (although this is merely a suggestion rather than a prescription, as the display process is outside the scope of this document), the source scan type of the associated picture, and whether the associated picture is a duplicate of a previous picture, in output order, of the same layer.

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

- A fixed picture rate indicator associated with a temporal sublayer, denoted herein by FixedPicRateWithinCvsFlag, such that value 1 indicates that the temporal distance between the display times of consecutive pictures in output order is constrained and value 0 indicates no such constraint.
- A display elemental period indicator, denoted herein by DisplayElementalPeriods, that indicates the number of elemental picture period intervals that the current coded picture occupies for the display model.

ffi_field_pic_flag equal to 1 indicates that the display model considers the current picture as a field, and **ffi_field_pic_flag** equal to 0 indicates that the display model considers the current picture as a frame.

ffi_bottom_field_flag equal to 1, when present, indicates that the current picture is a bottom field (i.e., that the parity of the current picture is bottom). **ffi_bottom_field_flag** equal to 0 indicates that the current picture is a top field (i.e., that the parity of the current picture is top). The two parities, bottom and top, are considered as opposite parities.

ffi_pairing_indicated_flag equal to 1, when present, indicates that the current picture is considered paired with the next picture in output order or with the previous picture in output order as the two fields of a frame. **ffi_pairing_indicated_flag** equal to 0, when present, indicates that a pairing of the current picture with another picture to form a frame is not expressed.

ffi_paired_with_next_field_flag equal to 1, when present, indicates that the current picture is considered paired with the next picture as the two fields of a frame. **ffi_paired_with_next_field_flag** equal to 0, when present, indicates that the current picture is considered paired with the previous picture as the two fields of a frame.

When **ffi_paired_with_next_field_flag** is present, the following constraints shall apply

- If **ffi_paired_with_next_field_flag** is equal to 0, there shall be at least one picture in the CLVS that precedes the current picture in output order and the picture that precedes the current picture in output order shall have the opposite parity and **ffi_pairing_indicated_flag** equal to 1 and the value of **ffi_paired_with_next_field_flag** for that preceding picture in output order shall be equal to 1.
- Otherwise, there shall be at least one picture in the CLVS that follows the current picture in output order and the picture that follows the current picture in output order shall have the opposite parity and **ffi_pairing_indicated_flag** equal to 1 and the value of **ffi_paired_with_next_field_flag** for that following picture in output order shall be equal to 0.

ffi_display_fields_from_frame_flag equal to 1, when present, indicates that the display model operates by sequentially displaying the individual fields of the frame with alternating parity. **ffi_display_fields_from_frame_flag** equal to 0, when present, indicates that the display model operates by displaying the current picture as a complete frame.

ffi_top_field_first_flag equal to 1, when present, indicates that the first field of the frame that is displayed by the display model is the top field. **ffi_top_field_first_flag** equal to 0, when present, indicates that the first field of the frame that is displayed by the display model is the bottom field.

ffi_display_elemental_periods_minus1 plus 1, when present, indicates the number of elemental picture period intervals that the current coded picture or field occupies for the display model. The value of **ffi_display_elemental_periods_minus1** shall be equal to DisplayElementalPeriods – 1.

The interpretation of combinations of **ffi_field_pic_flag**, FixedPicRateWithinCvsFlag, **ffi_bottom_field_flag**, **ffi_display_fields_from_frame_flag**, **ffi_top_field_first_flag**, and **ffi_display_elemental_periods_minus1** (through DisplayElementalPeriods) is specified in [Table 14](#), in which syntax elements that are not present are indicated by "-". Combinations of syntax elements that are not listed in [Table 14](#) are

reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this edition of this document.

NOTE 1 When FixedPicRateWithinCvsFlag is equal to 1, the indicated display times are constrained to account for time duration for a display model that follows the display patterns indicated by the values of the syntax elements of the frame-field information SEI message (although the display process is outside the scope of this document). Although the video decoder model might be specified to only output the entire cropped decoded picture, the modelled display behaviour sometimes includes other steps, such as the repeated display of a frame for multiple time intervals when ffi_display_fields_from_frame_flag is equal to 0 or the sequential display of the individual fields of a frame when ffi_display_fields_from_frame_flag is equal to 1.

NOTE 2 Frame doubling can be used to facilitate the display, for example, of 25 Hz progressive-scan video on a 50 Hz progressive-scan display or 30 Hz progressive-scan video on a 60 Hz progressive-scan display. Using frame doubling and frame tripling in alternating combination on every other frame could be used to facilitate the display of 24 Hz progressive-scan video on a 60 Hz progressive-scan display.

Table 14 — Interpretation of frame-field information syntax elements

ffi_field_pic_flag	FixedPicRateWithinCvsFlag	ffi_bottom_field_flag	ffi_display_fields_from_frame_flag	ffi_top_field_first_flag	DisplayElementalPeriods	Indicated display of the picture by the display model
0	0	-	0	-	1	(progressive) Frame
		-	1	0	2	Bottom field, top field, in that order
		-	1	1	2	Top field, bottom field, in that order
		-	1	0	3	Bottom field, top field, bottom field repeated, in that order
		-	1	1	3	Top field, bottom field, top field repeated, in that order
	1	-	0	-	n	(progressive) Frame displayed for n elemental periods of time
		-	1	0	2	Bottom field, top field, in that order, each displayed for 1 elemental period of time
		-	1	1	2	Top field, bottom field, in that order, each displayed for 1 elemental period of time
		-	1	0	3	Bottom field, top field, bottom field repeated, in that order, each displayed for 1 elemental period of time
		-	1	1	3	Top field, bottom field, top field repeated, in that order, each displayed for 1 elemental period of time
1	0	0	-	-	1	Top field
		1	-	-	1	Bottom field
	1	0	-	-	1	Top field displayed for 1 elemental period of time
		1	-	-	-	Bottom field displayed for 1 elemental period of time

ffi_source_scan_type equal to 1 indicates that the source scan type of the associated picture should be interpreted as progressive. **ffi_source_scan_type** equal to 0 indicates that the source scan type of the associated picture should be interpreted as interlaced. **ffi_source_scan_type** equal to 2 indicates that the source scan type of the associated picture is unknown or unspecified or specified by other means not specified in this document. **ffi_source_scan_type** equal to 3 is reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this edition of this document. Decoders conforming to this edition of this document shall interpret the value 3 for **ffi_source_scan_type** as equivalent to the value 2.

ffi_duplicate_flag equal to 1 indicates that the current picture is indicated to be a duplicate of a previous picture in output order. **ffi_duplicate_flag** equal to 0 indicates that the current picture is not indicated to be a duplicate of a previous picture in output order.

NOTE 3 The **ffi_duplicate_flag** could be used to mark coded pictures known to have originated from a repetition process such as "3:2 pull-down" or other such duplication and picture rate interpolation methods. This flag would commonly be used when a video feed is encoded as a field sequence in a "transport pass-through" fashion, with known duplicate pictures tagged by setting **ffi_duplicate_flag** equal to 1.

NOTE 4 When **ffi_field_pic_flag** is equal to 1 and **ffi_duplicate_flag** is equal to 1, this could be interpreted as an indication that the AU contains a field that duplicates the content of the previous field in output order with the same parity as the current field.

8.17 Sample aspect ratio information SEI message

8.17.1 Sample aspect ratio information SEI message syntax

sample_aspect_ratio_info(payloadSize) {	Descriptor
sari_cancel_flag	u(1)
if(!sari_cancel_flag) {	
sari_persistence_flag	u(1)
sari_aspect_ratio_idc	u(8)
if(sari_aspect_ratio_idc == 255) {	
sari_sar_width	u(16)
sari_sar_height	u(16)
}	
}	
}	

8.17.2 Sample aspect ratio information SEI message semantics

The SARI SEI message provides information about the sample aspect ratio of the samples of the associated decoded pictures. When **vui_aspect_ratio_constant_flag** is equal to 1, there shall be no SARI SEI messages present in the CLVS.

sari_cancel_flag equal to 1 indicates that the SARI SEI message cancels the persistence of any previous SARI SEI messages in output order that applies to the current layer. **sari_cancel_flag** equal to 0 indicates that SARI follows.

sari_persistence_flag specifies the persistence of the SARI SEI message for the current layer.

sari_persistence_flag equal to 0 specifies that the SARI applies to the current decoded picture only.

sar_persistence_flag equal to 1 specifies that the SARI 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 SARI SEI message is output that follows the current picture in output order.

sari_aspect_ratio_idc, when not equal to 255, indicates the sample aspect ratio of the luma samples of the decoded output picture, with the same semantics as specified for the SampleAspectRatio parameter in Rec. ITU-T H.273 | ISO/IEC 23091-2. When the sari_aspect_ratio_idc syntax element is not present, the value of sari_aspect_ratio_idc is inferred to be equal to 0. Values of sari_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 sari_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.

sari_sar_width, when present, indicates the horizontal size of the sample aspect ratio (in an arbitrary unit).

sari_sar_height, when present, indicates the vertical size of the sample aspect ratio (in the same arbitrary unit as sari_sar_width).

When present, sari_sar_width and sari_sar_height shall be relatively prime or equal to 0. When sari_aspect_ratio_idc is equal to 0 or sari_sar_width is equal to 0 or sari_sar_height is equal to 0, the sample aspect ratio is unknown or unspecified or specified by other means not specified in this document.

8.18 Annotated regions SEI message

8.18.1 Annotated regions SEI message syntax

	Descriptor
annotated_regions(payloadSize) {	
ar_cancel_flag	u(1)
if(!ar_cancel_flag) {	
ar_not_optimized_for_viewing_flag	u(1)
ar_true_motion_flag	u(1)
ar_occluded_object_flag	u(1)
ar_partial_object_flag_present_flag	u(1)
ar_object_label_present_flag	u(1)
ar_object_confidence_info_present_flag	u(1)
if(ar_object_confidence_info_present_flag)	
ar_object_confidence_length_minus1	u(4)
if(ar_object_label_present_flag) {	
ar_object_label_language_present_flag	u(1)
if(ar_object_label_language_present_flag) {	
while(!byte_aligned())	
ar_bit_equal_to_zero /* equal to 0 */	f(1)
ar_object_label_language	st(v)
}	
}	
ar_num_label_updates	ue(v)

8.18.2 Annotated regions SEI message semantics

The annotated regions SEI message carries parameters that identify annotated regions using bounding boxes representing the size and location of identified objects.

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

- A cropped picture width and picture height in units of luma samples, denoted herein by `CroppedWidth` and `CroppedHeight`, respectively
- A conformance cropping window left offset, `ConfWinLeftOffset`
- A conformance cropping window top offset, `ConfWinTopOffset`
- A chroma format indicator, denoted herein by `ChromaFormatIdc`, as described in [subclause 7.3](#).

The variables `SubWidthC` and `SubHeightC` are derived from `ChromaFormatIdc` as specified by [Table 2](#).

ar_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous annotated regions SEI message that is associated with one or more layers to which the annotated regions SEI message applies. `ar_cancel_flag` equal to 0 indicates that annotated regions information follows.

When `ar_cancel_flag` equal to 1 or a new CVS of the current layer begins, the variables `LabelAssigned[i]`, `ObjectTracked[i]`, and `ObjectBoundingBoxAvail` are set equal to 0 for `i` in the range of 0 to 255, inclusive.

ar_not_optimized_for_viewing_flag equal to 1 indicates that the decoded pictures that the annotated regions SEI message applies to are not optimized for user viewing, but rather are optimized for some other purpose such as algorithmic object classification performance. `ar_not_optimized_for_viewing_flag` equal to 0 indicates that the decoded pictures that the annotated regions SEI message applies to may or may not be optimized for user viewing.

ar_true_motion_flag equal to 1 indicates that the motion information in the coded pictures that the annotated regions SEI message applies to was selected with a goal of accurately representing object motion for objects in the annotated regions. `ar_true_motion_flag` equal to 0 indicates that the motion information in the coded pictures that the annotated regions SEI message applies to may or may not be selected with a goal of accurately representing object motion for objects in the annotated regions.

ar_occluded_object_flag equal to 1 indicates that the `ar_bounding_box_top[ar_object_idx[i]]`, `ar_bounding_box_left[ar_object_idx[i]]`, `ar_bounding_box_width[ar_object_idx[i]]`, and `ar_bounding_box_height[ar_object_idx[i]]` syntax elements represent the size and location of an object or a portion of an object that may not be visible or may be only partially visible within the cropped decoded picture. `ar_occluded_object_flag` equal to 0 indicates that the `ar_bounding_box_top[ar_object_idx[i]]`, `ar_bounding_box_left[ar_object_idx[i]]`, `ar_bounding_box_width[ar_object_idx[i]]`, and `ar_bounding_box_height[ar_object_idx[i]]` syntax elements represent the size and location of an object that is entirely visible within the cropped decoded picture. It is a requirement of bitstream conformance that the value of `ar_occluded_object_flag` shall be the same for all `annotated_regions()` syntax structures within a CVS.

ar_partial_object_flag_present_flag equal to 1 indicates that `ar_partial_object_flag[ar_object_idx[i]]` syntax elements are present. `ar_partial_object_flag_present_flag` equal to 0 indicates that `ar_partial_object_flag[ar_object_idx[i]]` syntax elements are not present. It is a requirement of bitstream conformance that the value of `ar_partial_object_flag_present_flag` shall be the same for all `annotated_regions()` syntax structures within a CVS.

ar_object_label_present_flag equal to 1 indicates that label information corresponding to objects in the annotated regions is present. `ar_object_label_present_flag` equal to 0 indicates that label information corresponding to the objects in the annotated regions is not present.

ar_object_confidence_info_present_flag equal to 1 indicates that `ar_object_confidence[ar_object_idx[i]]` syntax elements are present. `ar_object_confidence_info_present_flag` equal to 0 indicates that `ar_object_confidence[ar_object_idx[i]]` syntax elements are not present. It is a requirement of

bitstream conformance that the value of `ar_object_confidence_present_flag` shall be the same for all `annotated_regions()` syntax structures within a CVS.

`ar_object_confidence_length_minus1` plus 1 specifies the length, in bits, of the `ar_object_confidence[ar_object_idx[i]]` syntax elements. It is a requirement of bitstream conformance that the value of `ar_object_confidence_length_minus1` shall be the same for all `annotated_regions()` syntax structures within a CVS.

`ar_object_label_language_present_flag` equal to 1 indicates that the `ar_object_label_language` syntax element is present. `ar_object_label_language_present_flag` equal to 0 indicates that the `ar_object_label_language` syntax element is not present.

`ar_bit_equal_to_zero` shall be equal to zero.

`ar_object_label_language` contains a language tag as specified by IETF RFC 5646, followed by a null termination byte equal to 0x00. The length of the `ar_object_label_language` syntax element shall be less than or equal to 255 bytes, not including the null termination byte. When not present, the language of the label is unspecified.

`ar_num_label_updates` indicates the total number of labels associated with the annotated regions that are signalled. The value of `ar_num_label_updates` shall be in the range of 0 to 255, inclusive.

`ar_label_idx[i]` indicates the index of the signalled label. The value of `ar_label_idx[i]` shall be in the range of 0 to 255, inclusive.

`ar_label_cancel_flag` equal to 1 cancels the persistence scope of the `ar_label_idx[i]`-th label. `ar_label_cancel_flag` equal to 0 indicates that the `ar_label_idx[i]`-th label is assigned a signalled value.

`LabelAssigned[ar_label_idx[i]]` equal to 1 indicates that the `ar_label_idx[i]`-th label is assigned. `LabelAssigned[ar_label_idx[i]]` equal to 0 indicates that the `ar_label_idx[i]`-th label is not assigned.

`ar_label[ar_label_idx[i]]` specifies the contents of the `ar_label_idx[i]`-th label. The length of the `ar_label[ar_label_idx[i]]` syntax element shall be less than or equal to 255 bytes, not including the null termination byte.

`ar_num_object_updates` indicates the number of object updates to be signalled. `ar_num_object_updates` shall be in the range of 0 to 255, inclusive.

`ar_object_idx[i]` is the index of the object parameters to be signalled. `ar_object_idx[i]` shall be in the range of 0 to 255, inclusive.

`ar_object_cancel_flag` equal to 1 cancels the persistence scope of the `ar_object_idx[i]`-th object. `ar_object_cancel_flag` equal to 0 indicates that parameters associated with the `ar_object_idx[i]`-th object are signalled.

`ObjectTracked[ar_object_idx[i]]` equal to 1 indicates that the `ar_object_idx[i]`-th object is tracked. `ObjectTracked[ar_object_idx[i]]` equal to 0 indicates that the `ar_object_idx[i]`-th object is not tracked.

`ar_object_label_update_flag` equal to 1 indicates that an object label is signalled. `ar_object_label_update_flag` equal to 0 indicates that an object label is not signalled.

`ar_object_label_idx[ar_object_idx[i]]` indicates the index of the label corresponding to the `ar_object_idx[i]`-th object. When `ar_object_label_idx[ar_object_idx[i]]` is not present, its value is inferred from a previous annotated regions SEI message in output order in the same CVS, if any. The value of `ar_object_label_idx[ar_object_idx[i]]` shall be in the range of 0 to 255, inclusive.

`ar_bounding_box_update_flag` equal to 1 indicates that object bounding box parameters are signalled. `ar_bounding_box_update_flag` equal to 0 indicates that object bounding box parameters are not signalled.

`ar_bounding_box_cancel_flag` equal to 1 cancels the persistence scope of the `ar_bounding_box_top[ar_object_idx[i]]`, `ar_bounding_box_left[ar_object_idx[i]]`, `ar_bounding_box_width[ar_object_idx[i]]`, `ar_bounding_box_height[ar_object_idx[i]]`, `ar_bounding_box_right[ar_object_idx[i]]`, and `ar_bounding_box_bottom[ar_object_idx[i]]`.

$idx[i]$, $ar_bounding_box_height[ar_object_idx[i]]$, $ar_partial_object_flag[ar_object_idx[i]]$, and $ar_object_confidence[ar_object_idx[i]]$. $ar_bounding_box_cancel_flag$ equal to 0 indicates that $ar_bounding_box_top[ar_object_idx[i]]$, $ar_bounding_box_left[ar_object_idx[i]]$, $ar_bounding_box_width[ar_object_idx[i]]$, $ar_bounding_box_height[ar_object_idx[i]]$, $ar_partial_object_flag[ar_object_idx[i]]$, and $ar_object_confidence[ar_object_idx[i]]$ syntax elements are signalled.

$ObjectBoundingBoxAvail[ar_object_idx[i]]$ equal to 1 indicates that the bounding box information of the $object_idx[i]$ -th object is signalled. $ObjectBoundingBoxAvail[ar_object_idx[i]]$ equal to 0 indicates that the bounding box information of the $object_idx[i]$ -th object is not signalled.

$ar_bounding_box_top[ar_object_idx[i]]$, **$ar_bounding_box_left[ar_object_idx[i]]$** , **$ar_bounding_box_width[ar_object_idx[i]]$** , and **$ar_bounding_box_height[ar_object_idx[i]]$** specify the coordinates of the top-left corner and the width and height, respectively, of the bounding box of the $ar_object_idx[i]$ -th object in the cropped decoded picture, relative to the conformance cropping window specified by the active SPS.

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

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

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

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

The identified object rectangle contains the luma samples with horizontal picture coordinates from $SubWidthC * (ConfWinLeftOffset + ar_bounding_box_left[ar_object_idx[i]])$ to $SubWidthC * (ConfWinLeftOffset + ar_bounding_box_left[ar_object_idx[i]] + ar_bounding_box_width[ar_object_idx[i]]) - 1$, inclusive, and vertical picture coordinates from $SubHeightC * (ConfWinTopOffset + ar_bounding_box_top[ar_object_idx[i]])$ to $SubHeightC * (ConfWinTopOffset + ar_bounding_box_top[ar_object_idx[i]] + ar_bounding_box_height[ar_object_idx[i]]) - 1$, inclusive.

When $ChromaArrayType$ is not equal to 0, the corresponding specified samples of the two chroma arrays are the samples having picture coordinates $(x / SubWidthC, y / SubHeightC)$, where (x, y) are the picture coordinates of the specified luma samples.

The values of $ar_bounding_box_top[ar_object_idx[i]]$, $ar_bounding_box_left[ar_object_idx[i]]$, $ar_bounding_box_width[ar_object_idx[i]]$ and $ar_bounding_box_height[ar_object_idx[i]]$ persist in output order within the CVS for each value of $ar_object_idx[i]$. When not present, the values of $ar_bounding_box_top[ar_object_idx[i]]$, $ar_bounding_box_left[ar_object_idx[i]]$, $ar_bounding_box_width[ar_object_idx[i]]$ or $ar_bounding_box_height[ar_object_idx[i]]$ are inferred from a previous annotated regions SEI message in output order in the CVS, if any.

$ar_partial_object_flag[ar_object_idx[i]]$ equal to 1 indicates that the $ar_bounding_box_top[ar_object_idx[i]]$, $ar_bounding_box_left[ar_object_idx[i]]$, $ar_bounding_box_width[ar_object_idx[i]]$ and $ar_bounding_box_height[ar_object_idx[i]]$ syntax elements represent the size and location of an object that is only partially visible within the cropped decoded picture. $ar_partial_object_flag[ar_object_idx[i]]$ equal to 0 indicates that the $ar_bounding_box_top[ar_object_idx[i]]$, $ar_bounding_box_left[ar_object_idx[i]]$, $ar_bounding_box_width[ar_object_idx[i]]$ and $ar_bounding_box_height[ar_object_idx[i]]$ syntax elements represent the size and location of an object that may or may not be only partially visible within the cropped decoded picture. When not present, the value of $ar_partial_object_flag[ar_object_idx[i]]$ is inferred from a previous annotated regions SEI message in output order in the CVS, if any.

$ar_object_confidence[ar_object_idx[i]]$ indicates the degree of confidence associated with the $ar_object_idx[i]$ -th object, in units of $2^{-(ar_object_confidence_length_minus1 + 1)}$, such that a higher value of $ar_object_confidence[ar_object_idx[i]]$ indicates a higher degree of confidence. The length of the $ar_object_confidence[ar_object_idx[i]]$ syntax element is $ar_object_confidence_length_minus1 + 1$.

bits. When not present, the value of `_object_confidence[ar_object_idx[i]]` is inferred from a previous annotated regions SEI message in output order in the CVS, if any.

8.19 Scalability dimension information SEI message

8.19.1 Scalability dimension information SEI message syntax

	Descriptor
<code>scalability_dimension_info(payloadSize) {</code>	
sdi_max_layers_minus1	u(6)
sdi_multiview_info_flag	u(1)
sdi_auxiliary_info_flag	u(1)
if(sdi_multiview_info_flag sdi_auxiliary_info_flag) {	
if(sdi_multiview_info_flag)	
sdi_view_id_len_minus1	u(4)
for(i = 0; i <= sdi_max_layers_minus1; i++) {	
sdi_layer_id[i]	u(6)
if(sdi_multiview_info_flag)	
sdi_view_id_val[i]	u(v)
if(sdi_auxiliary_info_flag)	
sdi_aux_id[i]	u(8)
if(sdi_aux_id[i] > 0) {	
sdi_num_associated_primary_layers_minus1[i]	u(6)
for(j = 0; j <= sdi_num_associated_primary_layers_minus1[i]; j++)	
sdi_associated_primary_layer_idx[i][j]	u(6)
}	
}	
}	
}	
}	

8.19.2 Scalability dimension information SEI message semantics

The scalability dimension information (SDI) SEI message provides the SDI for each layer in the current CVS, i.e., the CVS containing the SDI SEI message, such as 1) when there may be multiple views, the view ID of each layer; and 2) when there may be auxiliary information (such as depth or alpha) carried by one or more layers, the auxiliary ID of each layer.

When an SDI SEI message is present in any AU of a CVS, an SDI SEI message shall be present for the first AU of the CVS. All SDI SEI messages in a CVS shall have the same content.

sdi_max_layers_minus1 plus 1 indicates the maximum number of layers in the current CVS.

sdi_multiview_info_flag equal to 1 indicates that the current CVS may have multiple views and the `sdi_view_id_val[]` syntax elements are present in the SDI SEI message. **sdi_multiview_info_flag** equal to 0 indicates that the current CVS does not have multiple views and the `sdi_view_id_val[]` syntax elements are not present in the SDI SEI message.

sdi_auxiliary_info_flag equal to 1 indicates that one or more layers in the current CVS may be auxiliary layers, which carry auxiliary information, and the `sdi_aux_id[]` syntax elements are present in the SDI SEI message. **sdi_auxiliary_info_flag** equal to 0 indicates that the current CVS does not have an auxiliary layer and the `sdi_aux_id[]` syntax elements are not present in the SDI SEI message.