



BSI Standards Publication

Information technology — Coding of audio-visual objects

Part 10: Advanced video coding

National foreword

This British Standard is the UK implementation of ISO/IEC 14496-10:2022.

The UK participation in its preparation was entrusted to Technical Committee IST/37, Coding of picture, audio, multimedia and hypermedia information.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2022
Published by BSI Standards Limited 2022

ISBN 978 0 539 18815 8

ICS 35.040.40

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 December 2022.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

**Information technology — Coding of
audio-visual objects —**

**Part 10:
Advanced video coding**

*Technologies de l'information — Codage des objets audiovisuels —
Partie 10: Codage visuel avancé*





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

Foreword	vi
0 Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Abbreviated terms	28
5 Conventions	29
5.1 Arithmetic operators.....	29
5.2 Logical operators.....	30
5.3 Relational operators.....	30
5.4 Bit-wise operators	30
5.5 Assignment operators	31
5.6 Range notation.....	31
5.7 Mathematical functions	31
5.8 Order of operation precedence	32
5.9 Variables, syntax elements, and tables	33
5.10 Text description of logical operations	34
5.11 Processes	35
6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships	35
6.1 Bitstream formats	35
6.2 Source, decoded, and output picture formats	35
6.3 Spatial subdivision of pictures and slices	40
6.4 Inverse scanning processes and derivation processes for neighbours	41
6.4.1 Inverse macroblock scanning process	41
6.4.2 Inverse macroblock partition and sub-macroblock partition scanning process	42
6.4.3 Inverse 4x4 luma block scanning process	43
6.4.4 Inverse 4x4 Cb or Cr block scanning process for ChromaArrayType equal to 3	44
6.4.5 Inverse 8x8 luma block scanning process	44
6.4.6 Inverse 8x8 Cb or Cr block scanning process for ChromaArrayType equal to 3	44
6.4.7 Inverse 4x4 chroma block scanning process	44
6.4.8 Derivation process of the availability for macroblock addresses	45
6.4.9 Derivation process for neighbouring macroblock addresses and their availability	45
6.4.10 Derivation process for neighbouring macroblock addresses and their availability in MBAFF frames	45
6.4.11 Derivation processes for neighbouring macroblocks, blocks, and partitions	46
6.4.12 Derivation process for neighbouring locations	51
6.4.13 Derivation processes for block and partition indices.....	54
7 Syntax and semantics	55
7.1 Method of specifying syntax in tabular form	55
7.2 Specification of syntax functions, categories, and descriptors	56
7.3 Syntax in tabular form.....	58
7.3.1 NAL unit syntax	58
7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax	59
7.3.3 Slice header syntax	67
7.3.4 Slice data syntax	72
7.3.5 Macroblock layer syntax	73
7.4 Semantics	80
7.4.1 NAL unit semantics.....	80
7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics	90
7.4.3 Slice header semantics.....	104
7.4.4 Slice data semantics.....	115
7.4.5 Macroblock layer semantics	116
8 Decoding process	129
8.1 NAL unit decoding process	130
8.2 Slice decoding process	131
8.2.1 Decoding process for picture order count.....	131

8.2.2	Decoding process for macroblock to slice group map	135
8.2.3	Decoding process for slice data partitions	138
8.2.4	Decoding process for reference picture lists construction.....	139
8.2.5	Decoded reference picture marking process	146
8.3	Intra prediction process.....	150
8.3.1	Intra_4x4 prediction process for luma samples.....	151
8.3.2	Intra_8x8 prediction process for luma samples.....	157
8.3.3	Intra_16x16 prediction process for luma samples.....	164
8.3.4	Intra prediction process for chroma samples.....	167
8.3.5	Sample construction process for I_PCM macroblocks	171
8.4	Inter prediction process.....	172
8.4.1	Derivation process for motion vector components and reference indices	174
8.4.2	Decoding process for Inter prediction samples	187
8.4.3	Derivation process for prediction weights	196
8.5	Transform coefficient decoding process and picture construction process prior to deblocking filter process ..	198
8.5.1	Specification of transform decoding process for 4x4 luma residual blocks	198
8.5.2	Specification of transform decoding process for luma samples of Intra_16x16 macroblock prediction mode	199
8.5.3	Specification of transform decoding process for 8x8 luma residual blocks	200
8.5.4	Specification of transform decoding process for chroma samples	200
8.5.5	Specification of transform decoding process for chroma samples with ChromaArrayType equal to 3	202
8.5.6	Inverse scanning process for 4x4 transform coefficients and scaling lists.....	203
8.5.7	Inverse scanning process for 8x8 transform coefficients and scaling lists.....	204
8.5.8	Derivation process for chroma quantization parameters	205
8.5.9	Derivation process for scaling functions.....	206
8.5.10	Scaling and transformation process for DC transform coefficients for Intra_16x16 macroblock type	207
8.5.11	Scaling and transformation process for chroma DC transform coefficients	208
8.5.12	Scaling and transformation process for residual 4x4 blocks	210
8.5.13	Scaling and transformation process for residual 8x8 blocks	212
8.5.14	Picture construction process prior to deblocking filter process.....	216
8.5.15	Intra residual transform-bypass decoding process	218
8.6	Decoding process for P macroblocks in SP slices or SI macroblocks	218
8.6.1	SP decoding process for non-switching pictures.....	219
8.6.2	SP and SI slice decoding process for switching pictures	221
8.7	Deblocking filter process	223
8.7.1	Filtering process for block edges	227
8.7.2	Filtering process for a set of samples across a horizontal or vertical block edge	229
9	Parsing process	235
9.1	Parsing process for Exp-Golomb codes	235
9.1.1	Mapping process for signed Exp-Golomb codes	237
9.1.2	Mapping process for coded block pattern	237
9.2	CAVLC parsing process for transform coefficient levels	240
9.2.1	Parsing process for total number of non-zero transform coefficient levels and number of trailing ones ..	241
9.2.2	Parsing process for level information	244
9.2.3	Parsing process for run information	246
9.2.4	Combining level and run information	249
9.3	CABAC parsing process for slice data	249
9.3.1	Initialization process	250
9.3.2	Binarization process.....	274
9.3.3	Decoding process flow.....	283
9.3.4	Arithmetic encoding process.....	304
Annex A (normative)	Profiles and levels	311
Annex B (normative)	Byte stream format	334
Annex C (normative)	Hypothetical reference decoder	337
Annex D (normative)	Supplemental enhancement information	358
Annex E (normative)	Video usability information	449
Annex F (normative)	Scalable video coding	470
Annex G (normative)	Multiview video coding	695

Annex H (normative) Multiview and depth video coding	762
Annex I (normative) Multiview and depth video with enhanced non-base view coding	811
Bibliography	867

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 <http://patents.iec.ch>).

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

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

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T (as ITU-T H.264).

This tenth edition cancels and replaces the ninth edition (ISO/IEC 14496-10:2020), which has been technically revised.

The main changes are as follows:

- addition of annotated regions and shutter interval information supplemental enhancement information messages.

A list of all parts in the ISO/IEC 14496 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.

0 Introduction

0.1 Prologue

As the costs for both processing power and memory have reduced, network support for coded video data has diversified, and advances in video coding technology have progressed, the need has arisen for an industry standard for compressed video representation with substantially increased coding efficiency and enhanced robustness to network environments. Toward these ends the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) formed a Joint Video Team (JVT) in 2001 for development of a new Recommendation | International Standard. The standard has since been maintained and enhanced jointly by VCEG and MPEG.

0.2 Purpose

This Recommendation | International Standard was developed in response to the growing need for higher compression of moving pictures for various applications such as videoconferencing, digital storage media, television broadcasting, internet streaming, and communication. It is also designed to enable the use of the coded video representation in a flexible manner for a wide variety of network environments. The use of this Recommendation | International Standard allows motion video to be manipulated as a form of computer data and to be stored on various storage media, transmitted and received over existing and future networks and distributed on existing and future broadcasting channels.

0.3 Applications

This Recommendation | International Standard is designed to cover a broad range of applications for video content including but not limited to the following:

- CATV: cable TV on optical networks, copper, etc.
- DBS: direct broadcast satellite video services.
- DSL: digital subscriber line video services.
- DTTB: digital terrestrial television broadcasting.
- ISM: interactive storage media (optical disks, etc.).
- MMM: multimedia mailing.
- MSPN: multimedia services over packet networks.
- RTC: real-time conversational services (videoconferencing, videophone, etc.).
- RVS: remote video surveillance.
- SSM: serial storage media (digital VTR, etc.).

0.4 Publication and versions of this document

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 1 refers to the first approved version of this Recommendation | International Standard.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 2 refers to the integrated text containing the corrections specified in the first technical corrigendum.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 3 refers to the integrated text containing both the first technical corrigendum (2004) and the first amendment, which is referred to as the "Fidelity range extensions".

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 4 refers to the integrated text containing the first technical corrigendum (2004), the first amendment (the "Fidelity range extensions"), and an additional technical corrigendum (2005).

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 5 refers to the integrated version 4 text with its specification of the High 4:4:4 profile removed.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 6 refers to the integrated version 5 text after its amendment to support additional colour space indicators.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 7 refers to the integrated version 6 text after its amendment to define five new profiles intended primarily for professional applications (the High 10 Intra, High 4:2:2 Intra, High 4:4:4 Intra,

CAVLC 4:4:4 Intra, and High 4:4:4 Predictive profiles) and two new types of supplemental enhancement information (SEI) messages (the post-filter hint SEI message and the tone mapping information SEI message).

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 8 refers to the integrated version 7 text after its amendment to specify scalable video coding in three profiles (Scalable Baseline, Scalable High, and Scalable High Intra profiles).

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 9 refers to the integrated version 8 text after applying the corrections specified in a third technical corrigendum.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 10 refers to the integrated version 9 text after its amendment to specify a profile for multiview video coding (the Multiview High profile) and to define additional SEI messages.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 11 refers to the integrated version 10 text after its amendment to define a new profile (the Constrained Baseline profile) intended primarily to enable implementation of decoders supporting only the common subset of capabilities supported in various previously-specified profiles.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 12 refers to the integrated version 11 text after its amendment to define a new profile (the Stereo High profile) for two-view video coding with support of interlaced coding tools and to specify an additional SEI message specified as the frame packing arrangement SEI message. The changes for versions 11 and 12 were processed as a single amendment in the ISO/IEC approval process.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 13 refers to the integrated version 12 text with various minor corrections and clarifications as specified in a fourth technical corrigendum.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 14 refers to the integrated version 13 text after its amendment to define a new level (Level 5.2) supporting higher processing rates in terms of maximum macroblocks per second and a new profile (the Progressive High profile) to enable implementation of decoders supporting only the frame coding tools of the previously-specified High profile.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 15 refers to the integrated version 14 text with miscellaneous corrections and clarifications as specified in a fifth technical corrigendum.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 16 refers to the integrated version 15 text after its amendment to define three new profiles intended primarily for communication applications (the Constrained High, Scalable Constrained Baseline, and Scalable Constrained High profiles).

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 17 refers to the integrated version 16 text after its amendment to define additional supplemental enhancement information (SEI) message data, including the multiview view position SEI message, the display orientation SEI message, and two additional frame packing arrangement type indication values for the frame packing arrangement SEI message (the 2D content and tiled arrangement type indication values).

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 18 refers to the integrated version 17 text after its amendment to specify the coding of depth signals, including the specification of an additional profile, the Multiview Depth High profile.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 19 refers to the integrated version 18 text after incorporating a correction to the sub-bitstream extraction process for multiview video coding.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 20 refers to the integrated version 19 text after its amendment to specify the combined coding of video view and depth enhancement, including the specification of an additional profile, the Enhanced Multiview Depth High profile.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 21 refers to the integrated version 20 text after its amendment to specify additional colorimetry identifiers and an additional model type in the tone mapping information SEI message.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 22 refers to the integrated version 21 text after its amendment to specify multi-resolution frame-compatible (MFC) enhancement for stereoscopic video coding, including the specification of an additional profile, the MFC High profile.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 23 refers to the integrated version 22 text after its amendment to specify multi-resolution frame-compatible (MFC) stereoscopic video with depth maps, including the specification of an additional profile, the MFC Depth High profile, and the mastering display colour volume SEI message, additional colour-related video usability information codepoint identifiers, and miscellaneous minor corrections and clarifications.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 24 refers to the integrated version 23 text after its amendment to specify additional levels of decoder capability supporting larger picture sizes (Levels 6, 6.1, and 6.2), the green metadata SEI message, the alternative depth information SEI message, additional colour-related video usability information codepoint identifiers, and miscellaneous minor corrections and clarifications.

Rec. ITU-T H.264 | ISO/IEC 14496-10 version 25 refers to the integrated version 24 text after its amendment to specify the Progressive High 10 profile; support for additional colour-related indicators, including the hybrid log-gamma transfer characteristics indication, the alternative transfer characteristics SEI message, the IC_{TC} colour matrix transformation, chromaticity-derived constant luminance and non-constant luminance colour matrix coefficients, the colour remapping information SEI message, and miscellaneous minor corrections and clarifications.

Rec. ITU-T H.264 | ISO/IEC 14496-10 version 26 refers to the integrated version 25 text after its amendment to specify additional SEI messages for ambient viewing environment, content light level information, content colour volume, equirectangular projection, cubemap projection, sphere rotation, region-wise packing, omnidirectional viewport, SEI manifest, and SEI prefix indication, and miscellaneous minor corrections and clarifications.

Rec. ITU-T H.264 | ISO/IEC 14496-10 version 27 (the current document) refers to the integrated version 26 text after its amendment to specify additional SEI messages for annotated regions (through referencing to Rec. ITU-T H.274 | ISO/IEC 23002-7) and shutter interval information, and miscellaneous minor corrections and clarifications.

This edition corresponds in technical content to the fourteenth edition in ITU-T (approved in August 2021).

0.5 Profiles and levels

This document is designed to be generic in the sense that it serves a wide range of applications, bit rates, resolutions, qualities, and services. Applications should cover, among other things, digital storage media, television broadcasting and real-time communications. In the course of creating this document, various requirements from typical applications have been considered, necessary algorithmic elements have been developed, and these have been integrated into a single syntax. Hence, this document will facilitate video data interchange among different applications.

Considering the practicality of implementing the full syntax of this document, however, a limited number of subsets of the syntax are also stipulated by means of "profiles" and "levels". These and other related terms are formally defined in Clause 3.

A "profile" is a subset of the entire bitstream syntax that is specified by this document. Within the bounds imposed by the syntax of a given profile it is still possible to require a very large variation in the performance of encoders and decoders depending upon the values taken by syntax elements in the bitstream such as the specified size of the decoded pictures. In many applications, it is currently neither practical nor economic to implement a decoder capable of dealing with all hypothetical uses of the syntax within a particular profile.

In order to deal with this problem, "levels" are specified within each profile. A level is a specified set of constraints imposed on values of the syntax elements in the bitstream. These constraints may be simple limits on values. Alternatively they may take the form of constraints on arithmetic combinations of values (e.g., picture width multiplied by picture height multiplied by number of pictures decoded per second).

Coded video content conforming to this document uses a common syntax. In order to achieve a subset of the complete syntax, flags, parameters, and other syntax elements are included in the bitstream that signal the presence or absence of syntactic elements that occur later in the bitstream.

0.6 Overview of the design characteristics

0.6.1 General

The coded representation specified in the syntax is designed to enable a high compression capability for a desired image quality. With the exception of the transform bypass mode of operation for lossless coding in the High 4:4:4 Intra, CAVLC 4:4:4 Intra, and High 4:4:4 Predictive profiles, and the I_PCM mode of operation in all profiles, the algorithm is typically not lossless, as the exact source sample values are typically not preserved through the encoding and decoding processes. A number of techniques may be used to achieve highly efficient compression. Encoding algorithms (not specified in this document) may select between inter and intra coding for block-shaped regions of each picture. Inter coding uses motion vectors for block-based inter prediction to exploit temporal statistical dependencies between different pictures. Intra coding uses various spatial prediction modes to exploit spatial statistical dependencies in the source signal for a single picture. Motion vectors and intra prediction modes may be specified for a variety of block sizes in the picture. The prediction residual is then further compressed using a transform to remove spatial correlation inside the transform block before it is quantized, producing an irreversible process that typically discards less important visual information while forming a close approximation to the source samples. Finally, the motion vectors or intra prediction modes are

combined with the quantized transform coefficient information and encoded using either variable length coding or arithmetic coding.

Scalable video coding is specified in Annex F allowing the construction of bitstreams that contain sub-bitstreams that conform to this document. For temporal bitstream scalability, i.e., the presence of a sub-bitstream with a smaller temporal sampling rate than the bitstream, complete access units are removed from the bitstream when deriving the sub-bitstream. In this case, high-level syntax and inter prediction reference pictures in the bitstream are constructed accordingly. For spatial and quality bitstream scalability, i.e., the presence of a sub-bitstream with lower spatial resolution or quality than the bitstream, NAL units are removed from the bitstream when deriving the sub-bitstream. In this case, inter-layer prediction, i.e., the prediction of the higher spatial resolution or quality signal by data of the lower spatial resolution or quality signal, is typically used for efficient coding. Otherwise, the coding algorithm as described in the previous paragraph is used.

Multiview video coding is specified in Annex G allowing the construction of bitstreams that represent multiple views. Similar to scalable video coding, bitstreams that represent multiple views may also contain sub-bitstreams that conform to this document. For temporal bitstream scalability, i.e., the presence of a sub-bitstream with a smaller temporal sampling rate than the bitstream, complete access units are removed from the bitstream when deriving the sub-bitstream. In this case, high-level syntax and inter prediction reference pictures in the bitstream are constructed accordingly. For view bitstream scalability, i.e., the presence of a sub-bitstream with fewer views than the bitstream, NAL units are removed from the bitstream when deriving the sub-bitstream. In this case, inter-view prediction, i.e., the prediction of one view signal by data of another view signal, is typically used for efficient coding. Otherwise, the coding algorithm as described in the previous paragraph is used.

An extension of multiview video coding that additionally supports the inclusion of depth maps is specified in Annex H, allowing the construction of bitstreams that represent multiple views with corresponding depth views. In a similar manner as with the multiview video coding specified in Annex G, bitstreams encoded as specified in Annex H may also contain sub-bitstreams that conform to this document.

A multiview video coding extension with depth information is specified in Annex I. Sub-bitstreams consisting of a texture base view conform to this document, sub-bitstreams consisting of multiple texture views may also conform to Annex G of this document, and sub-bitstreams consisting of one or more texture views and one or more depth views may also conform to Annex H of this document. Enhanced texture view coding that utilizes the associated depth views and decoding processes for depth views are specified for this extension.

Rec. ITU-T H.274 | ISO/IEC 23002-7 specifies the syntax and semantics of some video usability information (VUI) parameters and supplemental enhancement information (SEI) messages that do not affect the conformance specifications in Annex C and subclauses F.8, G.8, H.8, and I.8. Among these SEI messages, those for which the syntax and semantics are not specified in this document may be used together with this document. SEI messages for which the syntax and semantics are specified in this document may always be used together with this document. To enable being used together with this document, the SEI payload type value of an SEI message specified in Rec. ITU-T H.274 | ISO/IEC 23002-7 needs to be specified in this document. For example, the SEI payload type value 202 (for the annotated regions SEI message) is specified in subclause D.1.1, while the syntax and semantics of that SEI message are specified in Rec. ITU-T H.274 | ISO/IEC 23002-7.

0.6.2 Predictive coding

Because of the conflicting requirements of random access and highly efficient compression, two main coding types are specified. Intra coding is done without reference to other pictures. Intra coding may provide access points to the coded sequence where decoding can begin and continue correctly, but typically also shows only moderate compression efficiency. Inter coding (predictive or bi-predictive) is more efficient using inter prediction of each block of sample values from some previously decoded picture selected by the encoder. In contrast to some other video coding standards, pictures coded using bi-predictive inter prediction may also be used as references for inter coding of other pictures.

The application of the three coding types to pictures in a sequence is flexible, and the order of the decoding process is generally not the same as the order of the source picture capture process in the encoder or the output order from the decoder for display. The choice is left to the encoder and will depend on the requirements of the application. The decoding order is specified such that the decoding of pictures that use inter-picture prediction follows later in decoding order than other pictures that are referenced in the decoding process.

0.6.3 Coding of progressive and interlaced video

This document specifies a syntax and decoding process for video that originated in either progressive-scan or interlaced-scan form, which may be mixed together in the same sequence. The two fields of an interlaced frame are separated in capture time while the two fields of a progressive frame share the same capture time. Each field may be coded separately or the two fields may be coded together as a frame. Progressive frames are typically coded as a frame. For interlaced

video, the encoder can choose between frame coding and field coding. Frame coding or field coding can be adaptively selected on a picture-by-picture basis and also on a more localized basis within a coded frame. Frame coding is typically preferred when the video scene contains significant detail with limited motion. Field coding typically works better when there is fast picture-to-picture motion.

0.6.4 Picture partitioning into macroblocks and smaller partitions

As in previous video coding Recommendations and International Standards, a macroblock, consisting of a 16x16 block of luma samples and two corresponding blocks of chroma samples, is used as the basic processing unit of the video decoding process.

A macroblock can be further partitioned for inter prediction. The selection of the size of inter prediction partitions is a result of a trade-off between the coding gain provided by using motion compensation with smaller blocks and the quantity of data needed to represent the data for motion compensation. In this document the inter prediction process can form segmentations for motion representation as small as 4x4 luma samples in size, using motion vector accuracy of one-quarter of the luma sample grid spacing displacement. The process for inter prediction of a sample block can also involve the selection of the picture to be used as the reference picture from a number of stored previously-decoded pictures. Motion vectors are encoded differentially with respect to predicted values formed from nearby encoded motion vectors.

Typically, the encoder calculates appropriate motion vectors and other data elements represented in the video data stream. This motion estimation process in the encoder and the selection of whether to use inter prediction for the representation of each region of the video content is not specified in this document.

0.6.5 Spatial redundancy reduction

Both source pictures and prediction residuals have high spatial redundancy. This document is based on the use of a block-based transform method for spatial redundancy removal. After inter prediction from previously-decoded samples in other pictures or spatial-based prediction from previously-decoded samples within the current picture, the resulting prediction residual is split into 4x4 blocks. These are converted into the transform domain where they are quantized. After quantization many of the transform coefficients are zero or have low amplitude and can thus be represented with a small amount of encoded data. The processes of transformation and quantization in the encoder are not specified in this document.

0.7 How to read this document

It is suggested that the reader starts with Clause 1 (Scope) and moves on to Clause 3 (Terms and Terms and definitions). Clause 6 should be read for the geometrical relationship of the source, input, and output of the decoder. Clause 7 (Syntax and semantics) specifies the order to parse syntax elements from the bitstream. See subclauses 7.1 to 7.3 for syntactical order and see subclause 7.4 for semantics; i.e., the scope, restrictions, and conditions that are imposed on the syntax elements. The actual parsing for most syntax elements is specified in Clause 9 (Parsing process). Finally, Clause 8 (Decoding process) specifies how the syntax elements are mapped into decoded samples. Throughout reading this document, the reader should refer to Clauses 2 (Normative references), 4 (Abbreviated terms), and 5 (Conventions) as needed. Annexes A through F also form an integral part of this document.

Annex A specifies fourteen profiles (Baseline, Constrained Baseline, Main, Extended, High, Progressive High, Constrained High, High 10, High 4:2:2, High 4:4:4 Predictive, High 10 Intra, High 4:2:2 Intra, High 4:4:4 Intra, and CAVLC 4:4:4 Intra), each being tailored to certain application domains, and defines the so-called levels of the profiles. Annex B specifies syntax and semantics of a byte stream format for delivery of coded video as an ordered stream of bytes. Annex C specifies the hypothetical reference decoder and its use to check bitstream and decoder conformance. Annex D specifies syntax and semantics for supplemental enhancement information message payloads. Annex E specifies syntax and semantics of the video usability information parameters of the sequence parameter set.

Annex F specifies scalable video coding (SVC). The reader is referred to Annex F for the entire decoding process for SVC, which is specified there with references being made to Clauses 2 to 9 and Annexes A to E. Subclause F.10 specifies five profiles for SVC (Scalable Baseline, Scalable Constrained Baseline, Scalable High, Scalable Constrained High, and Scalable High Intra).

Annex F specifies multiview video coding (MVC) and multi-resolution frame compatible stereo coding (MFC). The reader is referred to Annex F for the entire decoding process for MVC and MFC, which is specified there with references being made to Clauses 2 to 9 and Annexes A to E. Subclause G.10 specifies two profiles for MVC (Multiview High and Stereo High) and one profile for MFC (MFC High).

Annex G specifies MVC extensions for inclusion of depth maps, referred to as multiview video coding with depth (MVCD). The reader is referred to Annex G for the entire decoding process for MVCD, which is specified there with references being made to Clauses 2 to 9 and Annexes A to E and Annex F. Subclause H.10 specifies two profiles for MVCD (Multiview Depth High and MFC Depth High).

Annex H specifies a multiview video coding extension with depth information (3D-AVC). The reader is referred to Annex H for the entire decoding process for 3D-AVC, which is specified there with references being made to Clauses 2 to 9 and Annexes A to E and Annexes F to G. Subclause I.10 specifies one profile for 3D-AVC.

Rec. ITU-T H.274 | ISO/IEC 23002-7 specifies the syntax and semantics of some video usability information (VUI) parameters and supplemental enhancement information (SEI) messages that do not affect the conformance specifications in Annex C and subclauses F.8, G.8, H.8, and I.8. Among these SEI messages, those for which the syntax and semantics are not specified in this document may be used together with this document.

0.8 Patent declarations

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

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have 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 statements of the holders of these patent rights are registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

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.

Information technology — Coding of audio-visual objects —

Part 10:

Advanced video coding

1 Scope

This document specifies advanced video coding for coding of audio-visual objects.

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.

Rec. ITU-T H.274 | ISO/IEC 23002-7, *Versatile supplemental enhancement information messages for coded video bitstreams*

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

ISO/IEC 10646, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

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

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

ISO 12232, *Photography – Digital still cameras – Determination of exposure index, ISO speed ratings, standard output sensitivity, and recommended exposure index*

ISO/IEC 23001-11, *Information technology — MPEG systems technologies — Part 11: Energy-efficient media consumption (green metadata)*

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

3 Terms and definitions

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

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

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

3.1 General terms related to advanced video coding

3.1.1

access unit

set of *NAL units* that are consecutive in *decoding order* and contain exactly one *primary coded picture*

Note 1 to entry: In addition to the primary coded picture, an access unit may also contain one or more redundant coded pictures, one auxiliary coded picture, or other *NAL units* not containing slices or slice data partitions of a coded picture. The decoding of an access unit always results in a decoded picture.

3.1.2

AC transform coefficient

any *transform coefficient* for which the *frequency index* in one or both dimensions is non-zero

3.1.3

adaptive binary arithmetic decoding process

entropy *decoding process* that derives the values of *bins* from a *bitstream* produced by an *adaptive binary arithmetic encoding process*

3.1.4

adaptive binary arithmetic encoding process

entropy *encoding process* that codes a sequence of *bins* and produces a *bitstream* that can be decoded using the *adaptive binary arithmetic decoding process*

Note 1 to entry: The encoding process is not specified in this document.

3.1.5

alpha blending

process in which an *auxiliary coded picture* is used in combination with a *primary coded picture* and with other data in the *display process*

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

Note 2 to entry: The alpha blending process and other data are not specified by this document.

3.1.6

arbitrary slice order

ASO

decoding order of slices in which the *macroblock address* of the first *macroblock* of some *slice* of a *slice group* may be less than the *macroblock address* of the first *macroblock* of some other preceding *slice* of the same *slice group* or, in the case of a *picture* that is coded using three separate colour planes, some other preceding *slice* of the same *slice group* within the same colour plane, or in which the *slices* of a *slice group* of a *picture* may be interleaved with the *slices* of one or more other *slice groups* of the *picture* or, in the case of a *picture* that is coded using three separate colour planes, with the *slices* of one or more other *slice groups* within the same colour plane

3.1.7

auxiliary coded picture

picture that supplements the *primary coded picture* that may be used in combination with other data in the *display process*

Note 1 to entry: An auxiliary coded picture has the same syntactic and semantic restrictions as a monochrome redundant coded picture. An auxiliary coded picture shall contain the same number of macroblocks as the primary coded picture. Auxiliary coded pictures have no normative effect on the decoding process. See also primary coded picture and redundant coded picture.

Note 2 to entry: The other data is not specified by this document.

3.1.8

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.

3.1.9

B slice

bi-predictive slice

slice that may be decoded using *intra prediction* or *inter prediction* using at most two *motion vectors* and *reference indices* to *predict* the sample values of each *block*

3.1.10

bin

one bit of a *bin string*

3.1.11

binarization

set of *bin strings* for all possible values of a *syntax element*

3.1.12

binarization process

unique mapping process of all possible values of a *syntax element* onto a set of *bin strings*

3.1.13**bin string
string of bins**

intermediate binary representation of values of *syntax elements* from the *binarization* of the *syntax element*

3.1.14**bitstream**

sequence of bits that forms the representation of *coded pictures* and associated data forming one or more *coded video sequences*

Note 1 to entry: Bitstream is a collective term used to refer either to a NAL unit stream or a byte stream.

3.1.15**block**

MxN (M-column by N-row) array of samples, or an MxN array of *transform coefficients*

3.1.16**bottom field**

one of two *fields* that comprise a *frame*

Note 1 to entry: Each row of a bottom field is spatially located immediately below a corresponding row of a top field.

3.1.17**bottom macroblock**

macroblock within a *macroblock pair* that contains the samples in the bottom row of samples for the *macroblock pair*

Note 1 to entry: For a field macroblock pair, the bottom macroblock represents the samples from the region of the bottom field of the frame that lie within the spatial region of the macroblock pair. For a frame macroblock pair, the bottom macroblock represents the samples of the frame that lie within the bottom half of the spatial region of the macroblock pair.

3.1.18**broken link**

location in a *bitstream* at which it is indicated that some subsequent *pictures* in *decoding order* may contain serious visual artefacts due to unspecified operations performed in the generation of the *bitstream*

3.1.19**byte**

sequence of 8 bits, written and read with the most significant bit on the left and the least significant bit on the right

Note 1 to entry: When represented in a sequence of data bits, the most significant bit of a byte is first.

3.1.20**byte-aligned**

position in a *bitstream* is byte-aligned when the position is an integer multiple of 8 bits from the position of the first bit in the *bitstream*

Note 1 to entry: A bit or byte or syntax element is said to be byte-aligned when the position at which it appears in a bitstream is byte-aligned.

3.1.21**byte stream**

encapsulation of a *NAL unit stream* containing *start code prefixes* and *NAL units*

Note 1 to entry: As specified in Annex B.

3.1.22**category**

number associated with each *syntax element*

Note 1 to entry: The category is used to specify the allocation of syntax elements to NAL units for slice data partitioning. It may also be used in a manner determined by the application to refer to classes of syntax elements in a manner not specified in this document.

3.1.23

chroma

adjective specifying that a sample array or single sample is representing one of the two colour difference signals related to the primary colours

Note 1 to entry: The symbols used for a chroma array or sample are Cb and Cr.

Note 2 to entry: The term chroma is used rather than the term chrominance in order to avoid the implication of the use of linear light transfer characteristics that is often associated with the term chrominance.

3.1.24

coded field

coded representation of a field

3.1.25

coded frame

coded representation of a frame

3.1.26

coded picture

coded representation of a picture

Note 1 to entry: A coded picture may be either a coded field or a coded frame. Coded picture is a collective term referring to a primary coded picture or a redundant coded picture, but not to both together.

3.1.27

coded picture buffer

CPB

first-in first-out buffer containing *access units* in *decoding order* specified in the *hypothetical reference decoder*

Note 1 to entry: The hypothetical reference decoder is specified in Annex C.

3.1.28

coded representation

data element as represented in its coded form

3.1.29

coded slice data partition NAL unit

NAL unit containing a *slice data partition*

3.1.30

coded slice NAL unit

NAL unit containing a *slice* that is not a *slice* of an *auxiliary coded picture*

3.1.31

coded video sequence

sequence of *access units* that consists, in decoding order, of an *IDR access unit* followed by zero or more non-IDR *access units* including all subsequent *access units* up to but not including any subsequent *IDR access unit*

3.1.32

component

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

3.1.33

complementary field pair

collective term for a *complementary reference field pair* or a *complementary non-reference field pair*

3.1.34

complementary non-reference field pair

two *non-reference fields* that are in consecutive *access units* in *decoding order* as two *coded fields* of opposite parity and share the same value of the *frame_num syntax element*, where the first *field* is not already a paired *field*

3.1.35

complementary reference field pair

two *reference fields* that are in consecutive *access units* in *decoding order* as two *coded fields* and share the same value of the *frame_num syntax element*, where the second *field* in *decoding order* is not an *IDR picture* and does not include a *memory_management_control_operation syntax element* equal to 5

3.1.36**constituent picture**

part of a spatially frame-packed stereoscopic video 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

3.1.37**context variable**

variable specified for the *adaptive binary arithmetic decoding process* of a *bin* by a formula containing recently decoded *bins*

3.1.38**DC transform coefficient**

transform coefficient for which the *frequency index* is zero in all dimensions

3.1.39**decoded picture**

picture derived by decoding a *coded picture*

Note 1 to entry: A decoded picture is either a decoded frame, or a decoded field. A decoded field is either a decoded top field or a decoded bottom field.

3.1.40**decoded picture buffer****DPB**

buffer holding *decoded pictures* for reference, output reordering, or output delay specified for the *hypothetical reference decoder*

Note 1 to entry: The hypothetical reference decoder is specified in Annex C.

3.1.41**decoder**

embodiment of a *decoding process*

3.1.42**decoder under test****DUT**

decoder that is tested for conformance by operating the *hypothetical stream scheduler* to deliver a conforming *bitstream* to the *decoder* and to the *hypothetical reference decoder* and comparing the values and timing of the output of the two *decoders*

3.1.43**decoding order**

order in which *syntax elements* are processed by the *decoding process*

3.1.44**decoding process**

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

3.1.45**direct prediction**

inter prediction for a *block* for which no *motion vector* is decoded

Note 1 to entry: Two direct prediction modes are specified that are referred to as spatial direct prediction and temporal prediction mode.

3.1.46**display process**

process having, as its input, the cropped *decoded pictures* that are the output of the *decoding process*

Note 1 to entry: The display process is not specified in this document.

3.1.47**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.

3.1.48

emulation prevention byte

byte equal to 0x03 that may be present within a *NAL unit*

Note 1 to entry: The presence of emulation prevention bytes ensures that no sequence of consecutive byte-aligned bytes in the NAL unit contains a start code prefix.

3.1.49

encoder

embodiment of an *encoding process*

3.1.50

encoding process

process that produces a *bitstream*

Note 1 to entry: A bitstream conforming to this document.

3.1.51

field

assembly of alternate rows of a *frame*

Note 1 to entry: A frame is composed of two fields, a top field and a bottom field.

3.1.52

field macroblock

macroblock containing samples from a single *field*

Note 1 to entry: All macroblocks of a coded field are field macroblocks. When macroblock-adaptive frame/field decoding is in use, some macroblocks of a coded frame may be field macroblocks.

3.1.53

field macroblock pair

macroblock pair decoded as two *field macroblocks*

3.1.54

field scan

specific sequential ordering of *transform coefficients* that differs from the *zig-zag scan* by scanning columns more rapidly than rows

Note 1 to entry: Field scan is used for transform coefficients in field macroblocks.

3.1.55

flag

variable that can take one of the two possible values 0 and 1

3.1.56

frame

frame contains an 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 frame consists of two fields, a top field and a bottom field.

3.1.57

frame macroblock

macroblock representing samples from the two *fields* of a *coded frame*

Note 1 to entry: When macroblock-adaptive frame/field decoding is not in use, all macroblocks of a coded frame are frame macroblocks. When macroblock-adaptive frame/field decoding is in use, some macroblocks of a coded frame may be frame macroblocks.

3.1.58

frame macroblock pair

macroblock pair decoded as two *frame macroblocks*

3.1.59

frequency index

one-dimensional or two-dimensional index associated with a *transform coefficient* prior to an *inverse transform* part of the *decoding process*

3.1.60**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 may 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.

3.1.61**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.

3.1.62**hypothetical reference decoder****HRD**

hypothetical *decoder* model that specifies constraints on the variability of conforming *NAL unit streams* or conforming *byte streams* that an encoding process may produce

3.1.63**hypothetical stream scheduler****HSS**

hypothetical delivery mechanism for the timing and data flow of the input of a *bitstream* into the *hypothetical reference decoder*

Note 1 to entry: The HSS is used for checking the conformance of a bitstream or a decoder.

3.1.64**I slice****intra slice**

slice that is not an *SI slice* that is decoded using *intra prediction* only

3.1.65**instantaneous decoding refresh access unit****IDR access unit**

access unit in which the *primary coded picture* is an *IDR picture*

3.1.66**instantaneous decoding refresh picture****IDR picture**

coded picture for which the variable *IdrPicFlag* is equal to 1

Note 1 to entry: An IDR picture causes the decoding process to mark all reference pictures as "unused for reference" immediately after the decoding of the IDR picture. All coded pictures that follow an IDR picture in decoding order can be decoded without inter prediction from any picture that precedes the IDR picture in decoding order. The first picture of each coded video sequence in decoding order is an IDR picture.

3.1.67**inter coding**

coding of a *block*, *macroblock*, *slice*, or *picture* that uses *inter prediction*

3.1.68**inter prediction**

prediction derived from decoded samples of *reference pictures* other than the current *decoded picture*

3.1.69**interpretation sample value**

possibly-altered value corresponding to a decoded sample value of an *auxiliary coded picture* that may be generated for use in the *display process*

Note 1 to entry: Interpretation sample values are not used in the decoding process and have no normative effect on the decoding process.

3.1.70**intra coding**

coding of a *block*, *macroblock*, *slice*, or *picture* that uses *intra prediction*

3.1.71

intra prediction

prediction derived from the decoded samples of the same decoded *slice*

3.1.72

inverse transform

part of the *decoding process* by which a set of *transform coefficients* are converted into spatial-domain values, or by which a set of *transform coefficients* are converted into *DC transform coefficients*

3.1.73

layer

one of a set of syntactical structures in a non-branching hierarchical relationship

Note 1 to entry: Higher layers contain lower layers. The coding layers are the coded video sequence, picture, slice, and macroblock layers.

3.1.74

level

defined set of constraints on the values that may be taken by the *syntax elements* and variables

Note 1 to entry: The same set of levels is defined for all profiles, with most aspects of the definition of each level being in common across different profiles. Individual implementations may, within specified constraints, support a different level for each supported profile. In a different context, a level is the value of a transform coefficient prior to scaling (see the definition of transform coefficient level).

3.1.75

list

one-dimensional array of *syntax elements* or variables

3.1.76

list 0 (list 1) motion vector

motion vector associated with a *reference index* pointing into *reference picture list 0 (list 1)*

3.1.77

list 0 (list 1) prediction

inter prediction of the content of a *slice* using a *reference index* pointing into *reference picture list 0 (list 1)*

3.1.78

local coordinate axes

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

3.1.79

luma

adjective specifying that a sample array or single sample is representing the monochrome signal related to the primary colours.

Note 1 to entry: The symbol or subscript used for luma is Y or L.

Note 2 to entry: The term luma is used rather than the term luminance in order to avoid the implication of 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.

3.1.80

macroblock

16x16 *block* of *luma* samples and two corresponding *blocks* of *chroma* samples of a *picture* that has three sample arrays, or a 16x16 *block* of samples of a monochrome *picture* or a *picture* that is coded using three separate colour planes

Note 1 to entry: The division of a *slice* or a *macroblock pair* into macroblocks is a *partitioning*.

3.1.81

macroblock-adaptive frame

macroblock-adaptive field decoding

decoding process for *coded frames* in which some *macroblocks* may be decoded as *frame macroblocks* and others may be decoded as *field macroblocks*

3.1.82**macroblock address**

<macroblock-adaptive frame not in use> index of a *macroblock* in a *macroblock raster scan* of the *picture* starting with zero for the top-left *macroblock* in a *picture*

Note 1 to entry: The macroblock address of the top macroblock of each macroblock pair is an even number and the macroblock address of the bottom macroblock of each macroblock pair is an odd number.

3.1.83**macroblock address**

<macroblock-adaptive frame in use> *top macroblock* of a *macroblock pair* is two times the index of the *macroblock pair* in a *macroblock pair raster scan* of the *picture*, and the *bottom macroblock* of a *macroblock pair* is the macroblock address of the corresponding *top macroblock* plus 1

Note 1 to entry: The macroblock address of the top macroblock of each macroblock pair is an even number and the macroblock address of the bottom macroblock of each macroblock pair is an odd number.

3.1.84**macroblock location**

two-dimensional coordinates of a *macroblock* in a *picture* denoted by (*x*, *y*)

Note 1 to entry: For the top left macroblock of the picture (*x*, *y*) is equal to (0, 0). *x* is incremented by 1 for each macroblock column from left to right. When macroblock-adaptive frame/field decoding is not in use, *y* is incremented by 1 for each macroblock row from top to bottom. When macroblock-adaptive frame/field decoding is in use, *y* is incremented by 2 for each macroblock pair row from top to bottom, and is incremented by an additional 1 when a macroblock is a bottom macroblock.

3.1.85**macroblock pair**

pair of vertically contiguous *macroblocks* in a *frame* that is coupled for use in *macroblock-adaptive frame/field decoding*

Note 1 to entry: The division of a slice into macroblock pairs is a partitioning.

3.1.86**macroblock partition**

block of *luma* samples and two corresponding *blocks* of *chroma* samples resulting from a *partitioning* of a *macroblock* for *inter prediction* for a *picture* that has three sample arrays or a *block* of *luma* samples resulting from a *partitioning* of a *macroblock* for *inter prediction* for a monochrome *picture* or a *picture* that is coded using three separate colour planes

3.1.87**macroblock to slice group map**

means of mapping *macroblocks* of a *picture* into *slice groups*

Note 1 to entry: The macroblock to slice group map consists of a list of numbers, one for each coded macroblock, specifying the slice group to which each coded macroblock belongs.

3.1.88**map unit to slice group map**

means of mapping *slice group map units* of a *picture* into *slice groups*

Note 1 to entry: The map unit to slice group map consists of a list of numbers, one for each slice group map unit, specifying the slice group to which each coded slice group map unit belongs.

3.1.89**matrix**

two-dimensional array of *syntax elements* or variables

3.1.90**memory management control operation**

seven operations that control *reference picture marking*

3.1.91**motion vector**

two-dimensional vector used for *inter prediction* that provides an offset from the coordinates in the *decoded picture* to the coordinates in a *reference picture*

3.1.92**NAL unit**

syntax structure containing an indication of the type of data to follow and *bytes* containing that data in the form of an *RBSP* interspersed as necessary with *emulation prevention bytes*

3.1.93

NAL unit stream

sequence of *NAL units*

3.1.94

non-paired field

collective term for a *non-paired reference field* or a *non-paired non-reference field*

3.1.95

non-paired non-reference field

decoded *non-reference field* that is not part of a *complementary non-reference field pair*

3.1.96

non-paired reference field

decoded *reference field* that is not part of a *complementary reference field pair*

3.1.97

non-reference field

field coded with *nal_ref_idc* equal to 0

3.1.98

non-reference frame

frame coded with *nal_ref_idc* equal to 0

3.1.99

non-reference picture

picture coded with *nal_ref_idc* equal to 0

Note 1 to entry: A non-reference picture is not used for inter prediction of any other pictures.

3.1.100

omnidirectional video

video content in a format that enables rendering according to the user's viewing orientation

EXAMPLE If viewed using a head-mounted device, or according to a user's desired viewport, reflecting a potentially rotated viewing position.

3.1.101

opposite parity

opposite parity of *top* is *bottom*, and vice versa

3.1.102

output order

order in which the *decoded pictures* are output from the *decoded picture buffer*

3.1.103

P slice

predictive slice

slice that is not an *SP slice* that may be decoded using *intra prediction* or *inter prediction* using at most one *motion vector* and *reference index* to *predict* the sample values of each *block*

3.1.104

packed region

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

3.1.105

parameter

syntax element of a *sequence parameter set* or a *picture parameter set*

Note 1 to entry: Parameter is also used as part of the defined term quantization parameter.

3.1.106

parity

< field> *top* or *bottom*

3.1.107

partitioning

division of a set into subsets such that each element of the set is in exactly one of the subsets

3.1.108**picture**

collective term for a *field* or a *frame*

3.1.109**picture parameter set**

syntax structure containing *syntax elements* that apply to zero or more entire *coded pictures* as determined by the *pic_parameter_set_id syntax element* found in each *slice header*

3.1.110**picture order count**

variable that is associated with each *coded field* and each *field* of a *coded frame* and has a value that is non-decreasing with increasing *field* position in *output order* relative to the first output *field* of the previous *IDR picture* in *decoding order* or relative to the first output *field* of the previous *picture*, in *decoding order*, that contains a *memory management control operation* that marks all *reference pictures* as "unused for reference"

3.1.111**prediction**

embodiment of the *prediction process*

3.1.112**prediction process**

use of a *predictor* to provide an estimate of the sample value or data element currently being decoded

3.1.113**predictor**

combination of specified values or previously decoded sample values or data elements used in the *decoding process* of subsequent sample values or data elements

3.1.114**primary coded picture**

coded representation of a *picture* to be used by the *decoding process* for a bitstream

Note 1 to entry: The primary coded picture contains all macroblocks of the picture. The only pictures that have a normative effect on the decoding process are primary coded pictures. See also redundant coded picture.

3.1.115**profile**

specified subset of syntax

Note 1 to entry: The syntax of this document.

3.1.116**projected picture**

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

3.1.117**projected region**

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

3.1.118**projection**

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

3.1.119**quantization parameter**

variable used by the *decoding process* for *scaling of transform coefficient levels*

3.1.120**random access**

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

3.1.121

raster scan

mapping of a rectangular two-dimensional pattern to a one-dimensional pattern such that the first entries in the one-dimensional pattern are from the first top row of the two-dimensional pattern scanned from left to right, followed similarly by the second, third, etc., rows of the pattern (going down) each scanned from left to right

3.1.122

raw byte sequence payload

RBSP

syntax structure containing an integer number of *bytes* that is encapsulated in a *NAL unit*

Note 1 to entry: An RBSP is either empty or has the form of a string of data bits containing syntax elements followed by an RBSP stop bit and followed by zero or more subsequent bits equal to 0.

3.1.123

raw byte sequence payload stop bit

RBSP stop bit

bit equal to 1 present within a *raw byte sequence payload* after a *string of data bits*

Note 1 to entry: The location of the end of the string of data bits within an RBSP can be identified by searching from the end of the RBSP for the RBSP stop bit, which is the last non-zero bit in the RBSP.

3.1.124

recovery point

point in the *bitstream* at which the recovery of an exact or an approximate representation of the *decoded pictures* represented by the *bitstream* is achieved after a *random access* or *broken link*

3.1.125

redundant coded picture

coded representation of a *picture* or a part of a *picture*

Note 1 to entry: The content of a redundant coded picture shall not be used by the decoding process for a bitstream conforming to this document. A redundant coded picture is not required to contain all macroblocks in the primary coded picture. Redundant coded pictures have no normative effect on the decoding process. See also primary coded picture.

3.1.126

reference field

field used for *inter prediction* when *P*, *SP*, and *B slices* of a *coded field* or *field macroblocks* of a *coded frame* are decoded

Note 1 to entry: See also reference picture.

3.1.127

reference frame

frame used for *inter prediction* when *P*, *SP*, and *B slices* of a *coded frame* are decoded

Note 1 to entry: See also reference picture.

3.1.128

reference index

index into a *reference picture list*

3.1.129

reference picture

picture with *nal_ref_idc* not equal to 0

Note 1 to entry: A reference picture contains samples that may be used for inter prediction in the decoding process of subsequent pictures in decoding order.

3.1.130

reference picture list

list of *reference pictures* that is used for *inter prediction* of a *P*, *B*, or *SP slice*

Note 1 to entry: For the decoding process of a *P* or *SP slice*, there is one reference picture list. For the decoding process of a *B slice*, there are two reference picture lists.

3.1.131

reference picture list 0

reference picture list used for *inter prediction* of a *P*, *B*, or *SP slice*

Note 1 to entry: All inter prediction used for P and SP slices uses reference picture list 0. Reference picture list 0 is one of two reference picture lists used for inter prediction for a B slice, with the other being reference picture list 1.

3.1.132

reference picture list 1

reference picture list used for *inter prediction* of a *B slice*

Note 1 to entry: Reference picture list 1 is one of two reference picture lists used for inter prediction for a B slice, with the other being reference picture list 0.

3.1.133

reference picture marking

Specifies, in the bitstream, how the *decoded pictures* are marked for *inter prediction*

3.1.134

region-wise packed picture

decoded picture that contains one or more *packed regions*

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

3.1.135

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*

3.1.136

reserved

<particular syntax element> values for future use by ITU-T | ISO/IEC

Note 1 to entry: These values shall not be used in bitstreams conforming to this document, but may be used in future extensions of this document by ITU-T | ISO/IEC.

3.1.137

residual

decoded difference between a *prediction* of a sample or data element and its decoded value

3.1.138

run

number of consecutive data elements represented in the decoding process

Note 1 to entry: In one context, the number of zero-valued transform coefficient levels preceding a non-zero transform coefficient level in the list of transform coefficient levels generated by a zig-zag scan or a field scan. In other contexts, run refers to a number of macroblocks.

3.1.139

sample aspect ratio

Specifies, for assisting the *display process*, which is not specified in this document, the ratio between the intended horizontal distance between the columns and the intended vertical distance between the rows of the *luma* sample array in a *frame*

Note 1 to entry: Sample aspect ratio is expressed as h:v, where h is horizontal width and v is vertical height (in arbitrary units of spatial distance).

3.1.140

scaling

process of multiplying *transform coefficient levels* by a factor, resulting in *transform coefficients*

3.1.141

sequence parameter set

syntax structure containing *syntax elements* that apply to zero or more entire *coded video sequences* as determined by the content of a *seq_parameter_set_id syntax element* found in the *picture parameter set* referred to by the *pic_parameter_set_id syntax element* found in each *slice header*

3.1.142

SI slice

switching I slice

slice that is coded using *intra prediction* only and using quantization of the *prediction samples*

Note 1 to entry: An SI slice can be coded such that its decoded samples can be constructed identically to an SP slice

3.1.143

skipped macroblock

macroblock for which no data is coded other than an indication that the *macroblock* is to be decoded as "skipped"

Note 1 to entry: This indication may be common to several macroblocks.

3.1.144

slice

integer number of *macroblocks* or *macroblock pairs* ordered consecutively in the *raster scan* within a particular *slice group*

Note 1 to entry: For the primary coded picture, the division of each slice group into slices is a partitioning. Although a slice contains macroblocks or macroblock pairs that are consecutive in the raster scan within a slice group, these macroblocks or macroblock pairs are not necessarily consecutive in the raster scan within the picture. The macroblock addresses are derived from the first macroblock address in a slice (as represented in the slice header) and the macroblock to slice group map, and, when a picture is coded using three separate colour planes, a colour plane identifier.

3.1.145

slice data partition

non-empty subset of the *syntax elements* of the slice data *syntax structure* for a *slice*

Note 1 to entry: The syntax elements of a slice data partition are associated with the same category.

3.1.146

slice data partitioning

method of *partitioning* selected *syntax elements* into *syntax structures* based on a *category* associated with each *syntax element*

3.1.147

slice group

subset of the *macroblocks* or *macroblock pairs* of a *picture*

Note 1 to entry: The division of the picture into slice groups is a partitioning of the picture. The partitioning is specified by the macroblock to slice group map.

3.1.148

slice group map units

units of the *map unit to slice group map*

3.1.149

slice header

part of a coded *slice* containing the data elements pertaining to the first or all *macroblocks* represented in the *slice*

3.1.150

source

video material or some of its attributes before encoding

3.1.151

SP slice

switching P slice

slice that may be coded using *intra prediction* or *inter prediction* with quantization of the *prediction* samples using at most one *motion vector* and *reference index* to *predict* the sample values of each *block*

Note 1 to entry: An SP slice can be coded such that its decoded samples can be constructed identically to another SP slice or an SI slice.

3.1.152

sphere coordinates

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

3.1.153

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

3.1.154

start code prefix

unique sequence of three *bytes* equal to 0x000001 embedded in the *byte stream* as a prefix to each *NAL unit*

Note 1 to entry: The location of a start code prefix can be used by a decoder to identify the beginning of a new NAL unit and the end of a previous NAL unit. Emulation of start code prefixes is prevented within NAL units by the inclusion of emulation prevention bytes.

3.1.155

string of data bits

SODB

sequence of some number of bits representing *syntax elements* present within a *raw byte sequence payload* prior to the *raw byte sequence payload stop bit*

Note 1 to entry: Within an SODB, 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.

3.1.156

sub-macroblock

one quarter of the samples of a *macroblock*, i.e., an 8x8 *luma block* and two corresponding *chroma blocks* of which one corner is located at a corner of the *macroblock* for a *picture* that has three sample arrays or an 8x8 *luma block* of which one corner is located at a corner of the *macroblock* for a monochrome *picture* or a *picture* that is coded using three separate colour planes

3.1.157

sub-macroblock partition

block of *luma* samples and two corresponding *blocks* of *chroma* samples resulting from a *partitioning* of a *sub-macroblock* for *inter prediction* for a *picture* that has three sample arrays or a *block* of *luma* samples resulting from a *partitioning* of a *sub-macroblock* for *inter prediction* for a monochrome *picture* or a *picture* that is coded using three separate colour planes

3.1.158

syntax element

element of data represented in the *bitstream*

3.1.159

syntax structure

zero or more *syntax elements* present together in the *bitstream* in a specified order

3.1.160

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

3.1.161

top field

one of two *fields* that comprise a *frame*

Note 1 to entry: Each row of a top field is spatially located immediately above the corresponding row of the bottom field.

3.1.162

top macroblock

macroblock within a *macroblock pair* that contains the samples in the top row of samples for the *macroblock pair*

Note 1 to entry: For a field macroblock pair, the top macroblock represents the samples from the region of the top field of the frame that lie within the spatial region of the macroblock pair. For a frame macroblock pair, the top macroblock represents the samples of the frame that lie within the top half of the spatial region of the macroblock pair.

3.1.163

transform coefficient

scalar quantity, considered to be in a frequency domain, that is associated with a particular one-dimensional or two-dimensional *frequency index* in an *inverse transform* part of the *decoding process*

3.1.164

transform coefficient level

integer quantity representing the value associated with a particular two-dimensional frequency index in the *decoding process* prior to *scaling* for computation of a *transform coefficient* value

3.1.165

universal unique identifier

UUID

identifier that is unique with respect to the space of all universal unique identifiers

3.1.166

unspecified

<particular syntax element> value which has no specified meaning

Note 1 to entry: These values will not have a specified meaning in the future as an integral part of this document.

3.1.167

variable length coding

VLC

reversible procedure for entropy coding that assigns shorter bit strings to *symbols* expected to be more frequent and longer bit strings to *symbols* expected to be less frequent

3.1.168

VCL NAL unit

collective term for *coded slice NAL units* and *coded slice data partition NAL units*

3.1.169

viewport

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

3.1.170

zig-zag scan

specific sequential ordering of *transform coefficient levels* from (approximately) the lowest spatial frequency to the highest

Note 1 to entry: Zig-zag scan is used for transform coefficient levels in frame macroblocks.

3.2 Terms related to scalable video coding (Annex F)

3.2.1

arbitrary slice order

ASO

decoding order of *slices* in which the *macroblock address* of the first *macroblock* of some *slice* of a *slice group* within a *layer representation* may be less than the *macroblock address* of the first *macroblock* of some other preceding *slice* of the same *slice group* within the same *layer representation* or in which the *slices* of a *slice group* within a *layer representation* may be interleaved with the *slices* of one or more other *slices groups* within the same *layer representation*

3.2.2

associated NAL unit

NAL unit that directly succeeds a *prefix NAL unit* in *decoding order*

3.2.3

B slice

slice that may be decoded using *intra-layer intra prediction* or *inter prediction* using at most two *motion vectors* and *reference indices* to *predict* the sample values of each *block*

3.2.4

base layer

bitstream subset that contains all *NAL units* with the *nal_unit_type syntax element* equal to 1 and 5 of the *bitstream* and does not contain any *NAL unit* with the *nal_unit_type syntax element* equal to 14, 15, or 20

Note 1 to entry: The bitstream subset also conforms to one or more of the profiles specified in Annex A.

3.2.5

base quality layer representation

layer representation of the *target dependency representation* of an *access unit* that is associated with the *quality_id syntax element* equal to 0

3.2.6

bitstream subset

bitstream that is derived as a *subset* from a *bitstream* by discarding zero or more *NAL units*

Note 1 to entry: A bitstream subset is also referred to as sub-bitstream.

3.2.7

bottom macroblock

macroblock within a *macroblock pair* that contains the samples in the bottom row of samples for the *macroblock pair*

Note 1 to entry: For a field macroblock pair, the bottom macroblock represents the samples from the region of the bottom field or layer bottom field of the frame or layer frame, respectively, that lie within the spatial region of the macroblock pair. For a frame macroblock pair, the bottom macroblock represents the samples of the frame or layer frame that lie within the bottom half of the spatial region of the macroblock pair.

3.2.8

coded slice in scalable extension NAL unit

coded slice NAL unit that contains an *EI slice*, *EP slice*, or an *EB slice*

3.2.9

complementary reference field pair

two *reference fields* that are in consecutive *access units* in *decoding order* as two *coded fields*, where the *target dependency representations* of the *fields* share the same value of the *frame_num syntax element* and where the second *field* in *decoding order* is not an *IDR picture* and the *target dependency representation* of the second *field* does not include a *memory_management_control_operation syntax element* equal to 5, or a *complementary reference base field pair*

3.2.10

complementary reference base field pair

Two *reference base fields* that are associated with two *coded fields* that are in consecutive *access units* in *decoding order*, where the *target dependency representations* of the *coded fields* share the same value of the *frame_num syntax element* and where the second *coded field* in *decoding order* is not an *IDR picture* and the *target dependency representation* of the second *coded field* does not include a *memory_management_control_operation syntax element* equal to 5

Note 1 to entry: A complementary reference base field pair is a complementary reference field pair.

3.2.11

dependency representation

subset of *VCL NAL units* within an *access unit* that are associated with the same value of the *dependency_id syntax element*, which is provided as part of the *NAL unit* header or by an associated *prefix NAL unit*, and the same value of the *redundant_pic_cnt syntax element*. A *dependency representation* consists of one or more *layer representations*

3.2.12

EB slice

slice that may be decoded using *intra prediction* or *inter prediction* or *inter-layer prediction* from *syntax elements* and derived variables of the *reference layer representation*

Note 1 to entry: For inter-prediction of EB slices at most two motion vectors and reference indices are used to predict the sample values of each block.

3.2.13

EI slice

slice that is not an *I slice* or *SI slice* that is decoded using *intra prediction* only

3.2.14

EP slice

slice that may be decoded using *intra prediction* or *inter prediction* or *inter-layer prediction* from *syntax elements* and derived variables of the *reference layer representation*

Note 1 to entry: For inter-prediction of EP slices at most one motion vector and reference index is used to predict the sample values of each block.

3.2.15

field macroblock

macroblock containing samples from a single *field* or *layer field*

3.2.16

frame macroblock

macroblock containing samples from the two *fields* or *layer fields* of a *frame* or *layer frame*, respectively

3.2.17

I slice

slice that is decoded using *intra-layer intra prediction* only

3.2.18

instantaneous decoding refresh picture

IDR picture

coded picture for which the variable *IdrPicFlag* is equal to 1 for the *target dependency representation*

Note 1 to entry: An IDR picture causes the decoding process to mark all reference pictures as "unused for reference" immediately after the decoding of the IDR picture. All coded pictures that follow an IDR picture in decoding order can be decoded without inter prediction from any picture that precedes the IDR picture in decoding order. The first picture of each coded video sequence in decoding order is an IDR picture.

3.2.19

inter-layer intra prediction

inter-layer prediction derived from decoded samples of *intra-coded macroblocks* of the *reference layer representation*

3.2.20

inter-layer prediction

prediction derived from *syntax elements*, derived variables, or decoded samples of the *reference layer representation*

3.2.21

intra-layer intra prediction

spatial intra prediction

prediction derived from decoded samples of the same decoded *slice*

3.2.22

intra prediction

collective term for *intra-layer intra prediction* or *inter-layer intra prediction* or a combination of *intra-layer intra prediction* together with *inter-layer prediction* from *syntax elements* and derived variables of the *reference layer representation*

3.2.23

intra slice

collective term for *I slice* or *EI slice*

3.2.24

layer bottom field

one of two *layer fields* that comprise a *layer frame*

Note 1 to entry: Each row of a layer bottom field is spatially located immediately below a corresponding row of a layer top field.

3.2.25

layer field

assembly of alternate rows of a *layer frame*

Note 1 to entry: A layer frame is composed of two layer fields, a layer top field and a layer bottom field.

3.2.26

layer frame

layer frame contains an array of *luma* samples that represents an intermediate decoding result for a *field* or a *frame* in monochrome format or an array of *luma* samples and two corresponding arrays of *chroma* samples that represent an intermediate decoding result for a *field* or a *frame* in 4:2:0, 4:2:2, and 4:4:4 colour format. A *layer frame* consists of two *layer fields*, a *layer top field* and a *layer bottom field*

3.2.27

layer picture

collective term for a *layer field* or a *layer frame*

3.2.28

layer top field

one of two *layer fields* that comprise a *layer frame*

Note 1 to entry: Each row of a layer top field is spatially located immediately above a corresponding row of a layer bottom field.

3.2.29

layer representation

subset of *VCL NAL units* within an *access unit* that are associated with the same values of the *dependency_id* and *quality_id syntax elements*, which are provided as part of the *VCL NAL unit header* or by an associated *prefix NAL unit*, and the same value of the *redundant_pic_cnt syntax element*. One or more *layer representations* represent a *dependency representation*

3.2.30

layer representation identifier

integer value by which a particular *layer representation* inside a *coded picture* is uniquely identified

3.2.31

macroblock

16x16 *block* of *luma* samples and two corresponding *blocks* of *chroma* samples of a *picture* or *layer picture* that has three sample arrays, or a 16x16 *block* of samples of a monochrome *picture* or *layer picture*

Note 1 to entry: The division of a slice or a macroblock pair into macroblocks is a partitioning.

3.2.32

macroblock-adaptive frame/field decoding

decoding process for *coded frames* or *layer representations* in which some *macroblocks* may be decoded as *frame macroblocks* and others may be decoded as *field macroblocks*

3.2.33

macroblock address

<macroblock-adaptive frame not in use> macroblock address is the index of a macroblock in a *macroblock raster scan* of the *picture* or *layer picture* starting with zero for the top-left *macroblock* in a *picture* or *layer picture*.

Note 1 to entry: The macroblock address of the top macroblock of each macroblock pair is an even number and the macroblock address of the bottom macroblock of each macroblock pair is an odd number.

3.2.34

macroblock address

<macroblock-adaptive frame in use> macroblock address of the *top macroblock* of a *macroblock pair* is two times the index of the *macroblock pair* in a *macroblock pair raster scan* of the *picture* or *layer picture*, and the macroblock address of the *bottom macroblock* of a *macroblock pair* is the macroblock address of the corresponding *top macroblock* plus 1

Note 1 to entry: The macroblock address of the top macroblock of each macroblock pair is an even number and the macroblock address of the bottom macroblock of each macroblock pair is an odd number.

3.2.35

macroblock location

two-dimensional coordinates of a *macroblock* in a *picture* or *layer picture* denoted by (*x*, *y*)

Note 1 to entry: For the top left macroblock of the picture or layer picture (*x*, *y*) is equal to (0, 0). *x* is incremented by 1 for each macroblock column from left to right. When macroblock-adaptive frame/field decoding is not in use, *y* is incremented by 1 for each macroblock row from top to bottom. When macroblock-adaptive frame/field decoding is in use, *y* is incremented by 2 for each macroblock pair row from top to bottom, and is incremented by an additional 1 when a macroblock is a bottom macroblock.

3.2.36

macroblock pair

pair of vertically contiguous *macroblocks* in a *frame* or *layer frame* that is coupled for use in *macroblock-adaptive frame/field decoding*

Note 1 to entry: The division of a slice into macroblock pairs is a partitioning.

3.2.37

macroblock to slice group map

means of mapping *macroblocks* of a *picture* or *layer picture* into *slice groups*

Note 1 to entry: The macroblock to slice group map consists of a list of numbers, one for each coded macroblock, specifying the slice group to which each coded macroblock belongs.

3.2.38

map unit to slice group map

means of mapping *slice group map units* of a *picture* or *layer picture* into *slice groups*

Note 1 to entry: The map unit to slice group map consists of a list of numbers, one for each slice group map unit, specifying the slice group to which each coded slice group map unit belongs to.

3.2.39

non-paired reference base field

reference base field that is not part of a *complementary reference base field pair*

Note 1 to entry: A non-paired reference base field is a non-paired reference field.

3.2.40

P slice

slice that may be decoded using *intra-layer intra prediction* or *inter prediction* using at most one *motion vector* and *reference index* to predict the sample values of each *block*

3.2.41

parameter

syntax element of an *SVC sequence parameter set* or a *picture parameter set*

Note 1 to entry: Parameter is also used as part of the defined term quantization parameter.

3.2.42

picture parameter set

syntax structure containing *syntax elements* that apply to zero or more *layer representations* as determined by the *pic_parameter_set_id syntax element* found in each *slice header*

3.2.43

prefix NAL unit

NAL unit with *nal_unit_type* equal to 14 that immediately precedes in *decoding order* a *NAL unit* with *nal_unit_type* equal to 1 or 5

Note 1 to entry: The NAL unit that immediately succeeds the prefix NAL unit in decoding order is referred to as the associated NAL unit. The prefix NAL unit contains data associated with the associated NAL unit, which are considered to be part of the associated NAL unit.

3.2.44

reference base field

reference field that is obtained by decoding a *base quality layer representation* with the *nal_ref_idc syntax element* not equal to 0, the *store_ref_base_pic_flag syntax element* equal to 1, and the *field_pic_flag syntax element* equal to 1 of a *coded picture* and all *layer representations* of the *coded picture* that are referred to by *inter-layer prediction* in the *base quality layer representation*

Note 1 to entry: A reference base field is not a decoded picture and it is not an output of the decoding process, but may be used for inter prediction when P, B, EP, and EB slices of a coded field or a field macroblock of a coded frame are decoded. See also reference base picture.

3.2.45

reference base frame

reference frame that is obtained by decoding a *base quality layer representation* with the *nal_ref_idc syntax element* not equal to 0, the *store_ref_base_pic_flag syntax element* equal to 1, and the *field_pic_flag syntax element* equal to 0 of a *coded picture* and all *layer representations* of the *coded picture* that are referred to by *inter-layer prediction* of the *base quality layer representation*

Note 1 to entry: A reference base frame is not a decoded picture and it is not an output of the decoding process, but may be used for inter prediction when P, B, EP, and EB slices of a coded frame are decoded. See also reference base picture.

3.2.46

reference base picture

collective term for a *reference base field* or a *reference base frame*

3.2.47

reference field

reference field may be used for *inter prediction* when P, B, EP, or EB slices of a *coded field* or *field macroblocks* of a *coded frame* are decoded

Note 1 to entry: See also reference picture.

3.2.48

reference frame

reference frame may be used for *inter prediction* when P, B, EP, or EB slices of a *coded frame* are decoded

Note 1 to entry: See also reference picture.

3.2.49**reference layer macroblock**

macroblock of a reference layer representation

3.2.50**reference layer representation**

reference layer representation for a particular *layer representation* of a *coded picture* is the *layer representation* that is used for *inter-layer prediction* of the particular *layer representation*

Note 1 to entry: The reference layer representation belongs to the same access unit as the layer representation that uses the reference layer representation for inter-layer prediction.

3.2.51**reference picture**

collective term for a *decoded picture* that is obtained by decoding a *coded picture* for which the *nal_ref_idc syntax element* that is associated with the *target dependency representation* is not equal to 0 or a *reference base picture*

Note 1 to entry: A reference picture contains samples that may be used for inter prediction in the decoding process of subsequent pictures in decoding order.

3.2.52**reference picture list**

list of *reference pictures* that is used for *inter prediction* of a *P*, *B*, *EP*, or *EB slice*

Note 1 to entry: For the decoding process of a *P* or *EP slice*, there is one reference picture list. For the decoding process of a *B* or *EB slice*, there are two reference picture lists.

3.2.53**reference picture list 0**

reference picture list used for *inter prediction* of a *P*, *B*, *EP*, or *EB slice*

Note 1 to entry: All inter prediction used for *P* and *EP slices* uses reference picture list 0. Reference picture list 0 is one of two reference picture lists used for inter prediction for a *B* or *EB slice*, with the other being reference picture list 1.

3.2.54**reference picture list 1**

reference picture list used for *inter prediction* of a *B* or *EB slice*

Note 1 to entry: Reference picture list 1 is one of two reference picture lists used for inter prediction for a *B* or *EB slice*, with the other being reference picture list 0.

3.2.55**scalable bitstream**

bitstream with the property that one or more *bitstream subsets* that are not identical to the scalable bitstream form another *bitstream* that conforms to this specification

3.2.56**sequence parameter set**

syntax structure containing *syntax elements* that apply to zero or more *layer representations* with the *dependency_id syntax element* equal to 0 and the *quality_id syntax element* equal to 0 as determined by the content of a *seq_parameter_set_id syntax element* found in the *picture parameter set* referred to by the *pic_parameter_set_id syntax element* found in each *slice header* of *I*, *P*, and *B slices*

3.2.57**slice**

integer number of *macroblocks* or *macroblock pairs* ordered consecutively in the *raster scan* within a particular *slice group*

Note 1 to entry: Each macroblock or macroblock pair of a picture or layer picture shall not be contained in more than one slice of a particular layer representation. Although a slice contains macroblocks or macroblock pairs that are consecutive in the raster scan within a slice group, these macroblocks or macroblock pairs are not necessarily consecutive in the raster scan within the picture or layer picture. The macroblock addresses are derived from the first macroblock address in a slice (as represented in the slice header) and the macroblock to slice group map.

3.2.58**slice group**

subset of the *macroblocks* or *macroblock pairs* of a *picture* or *layer picture*

Note 1 to entry: The division of the picture or layer picture into slice groups is a partitioning of the picture or layer picture. The partitioning is specified by the macroblock to slice group map.

3.2.59

sub-bitstream

subset of a bitstream

Note 1 to entry: A sub-bitstream is also referred to as bitstream subset.

3.2.60

subset

subset contains only elements that are also contained in the set from which the subset is derived

Note 1 to entry: The subset may be identical to the set from which it is derived.

3.2.61

subset sequence parameter set

syntax structure containing *syntax elements* that apply to zero or more *layer representations* with the *dependency_id syntax element* not equal to 0 or the *quality_id syntax element* not equal to 0 as determined by the content of a *seq_parameter_set_id syntax element* found in the *picture parameter set* referred to by the *pic_parameter_set_id syntax element* found in each *slice header* of *EI*, *EP*, and *EB slices*

3.2.62

SVC sequence parameter set

collective term for *sequence parameter set* or *subset sequence parameter set*

3.2.63

SVC sequence parameter set RBSP

collective term for *sequence parameter set RBSP* or *subset sequence parameter set RBSP*

3.2.64

target dependency representation

dependency representation of a *coded picture* that is associated with the largest value of the *dependency_id syntax element* for all *dependency representations* of the *coded picture*

3.2.65

target layer representation

layer representation of the *target dependency representation* of a *coded picture* that is associated with the largest value of the *quality_id syntax element* for all *layer representations* of the *target dependency representation* of the *coded picture*

3.2.66

top macroblock

macroblock within a *macroblock pair* that contains the samples in the top row of samples for the *macroblock pair*

Note 1 to entry: For a field macroblock pair, the top macroblock represents the samples from the region of the top field or layer top field of the frame or layer frame that lie within the spatial region of the macroblock pair. For a frame macroblock pair, the top macroblock represents the samples of the frame or layer frame that lie within the top half of the spatial region of the macroblock pair.

3.2.67

VCL NAL unit

collective term for *coded slice NAL units* and *prefix NAL units*

3.3 Terms related to multiview video coding (Annex G)

3.3.1

access unit

set of *NAL units* that are consecutive in *decoding order* and contain exactly one *primary coded picture* consisting of one or more *view components*

Note 1 to entry: In addition to the primary coded picture, an access unit may also contain one or more redundant coded pictures, one auxiliary coded picture, or other NAL units not containing slices or slice data partitions of a coded picture. The decoding of an access unit always results in one decoded picture consisting of one or more decoded view components.

3.3.2

anchor access unit

access unit in which the *primary coded picture* is an *anchor picture*

3.3.3**anchor picture**

coded picture in which all *slices* may reference only *slices* within the same *access unit*, i.e., *inter-view prediction* may be used, but no *inter prediction* is used, and all following *coded pictures* in output order do not use *inter prediction* from any *picture* prior to the *coded picture* in *decoding order*

Note 1 to entry: The value of *anchor_pic_flag* is equal to 1 for all the prefix NAL units (when present) and all the slice extension NAL units that are contained in an anchor picture.

3.3.4**anchor view component**

view component in an *anchor picture*

Note 1 to entry: All view components in an anchor picture are anchor view components.

3.3.5**associated NAL unit**

NAL unit that immediately follows a *prefix NAL unit* in *decoding order*

3.3.6**base view**

view that has the minimum value of *view order index* in a *coded video sequence*

Note 1 to entry: The base view can be decoded independently of other views, does not use inter-view prediction, and contains VCL NAL units only with *nal_unit_type* equal to 1, 5, or 14. The bitstream subset corresponding to the base view conforms to one or more of the profiles specified in Annex A. There is only one base view in a coded video sequence.

3.3.7**bitstream subset**

bitstream that is derived as a *subset* from a *bitstream* by discarding zero or more *NAL units*

Note 1 to entry: A bitstream subset is also referred to as a sub-bitstream.

3.3.8**coded slice MVC extension NAL unit**

coded slice NAL unit that has *nal_unit_type* equal to 20

3.3.9**decoded view component**

decoded view component is derived by decoding a *view component*

Note 1 to entry: A decoded view component is either a decoded frame view component, or a decoded field view component.

3.3.10**direct prediction**

inter prediction or *inter-view prediction* for a *block* for which no *motion vector* is decoded

Note 1 to entry: Two direct prediction modes are specified that are referred to as spatial direct prediction mode and temporal direct prediction mode.

3.3.11**field view component**

view component of a *field*

3.3.12**frame view component**

view component of a *frame*

3.3.13**instantaneous decoding refresh view component****IDR view component**

view component in an *IDR picture*

Note 1 to entry: All view components in an IDR picture are IDR view components. IDR view components are also anchor view components, and inter-view prediction may be used for IDR view components that are part of a non-base view.

3.3.14**inter-view coding**

coding of a *block*, *macroblock*, *slice*, or *picture* that uses *inter-view prediction*

3.3.15

inter-view only reference component

view component coded with *nal_ref_idc* equal to 0 and *inter_view_flag* equal to 1

Note 1 to entry: An inter-view only reference component contains samples that may be used for inter-view prediction in the decoding process of subsequent view components in decoding order, but are not used for inter prediction by any view components. Inter-view only reference components are non-reference pictures.

3.3.16

inter-view prediction

prediction derived from decoded samples of *inter-view reference components* or *inter-view only reference components* for decoding another *view component* in the same *access unit*

3.3.17

inter-view prediction reference

collective term for *inter-view reference components* or *inter-view only reference components*

3.3.18

inter-view reference component

view component coded with *nal_ref_idc* greater than 0 and *inter_view_flag* equal to 1

Note 1 to entry: An inter-view reference component contains samples that may be used for inter prediction of subsequent pictures in decoding order and inter-view prediction of subsequent view components in decoding order. Inter-view reference components are reference pictures.

3.3.19

left view

left part of a picture coded in a frame-packed manner with the side-by-side frame packing arrangement type or the top part of a picture coded in a frame-packed manner with the top-bottom frame packing arrangement type

3.3.20

list 0 (list 1) prediction

inter prediction or *inter-view prediction* of the content of a *slice* using a *reference index* pointing into *reference picture list 0 (list 1)*

3.3.21

macroblock partition

block of *luma* samples and two corresponding *blocks* of *chroma* samples resulting from a *partitioning* of a *macroblock* for *inter prediction* or *inter-view prediction*

3.3.22

motion vector

two-dimensional vector used for *inter prediction* or *inter-view prediction* that provides an offset from the coordinates in the *decoded view component* to the coordinates in a *reference picture* or *inter-view only reference component*

3.3.23

MVC sequence parameter set

collective term for *sequence parameter set* or *subset sequence parameter set*

3.3.24

MVC sequence parameter set RBSP

collective term for *sequence parameter set RBSP* or *subset sequence parameter set RBSP*

3.3.25

non-anchor access unit

access unit that is not an *anchor access unit*

3.3.26

non-anchor picture

coded picture that is not an *anchor picture*

3.3.27

non-anchor view component

view component that is not an *anchor view component*

3.3.28

non-base view

view that is not the *base view*

Note 1 to entry: VCL NAL units of a non-base view have `nal_unit_type` equal to 20. Decoding of view components in a non-base view may require the use of inter-view prediction.

3.3.29

non-reference picture

view component coded with `nal_ref_idc` equal to 0

Note 1 to entry: A non-reference picture is not used for inter prediction in the decoding process of any other view components.

3.3.30

operation point

operation point is identified by a `temporal_id` value representing the *target temporal level* and a set of `view_id` values representing the *target output views*

Note 1 to entry: One operation point is associated with a bitstream subset, which consists of the target output views and all other views the target output views depend on, that is derived using the sub-bitstream extraction process as specified in subclause G.4.5.3 with `IdTarget` equal to the `temporal_id` value and `viewIdTargetList` consisting of the set of `view_id` values as inputs. More than one operation point may be associated with the same bitstream subset. When the specification states "an operation point is decoded" it refers to the decoding of a bitstream subset corresponding to the operation point and subsequent output of the target output views.

3.3.31

picture order count

variable that is associated with each *field view component* and each *field* of a *frame view component* and has a value that is non-decreasing with increasing *field* position in *output order* in the same *view* relative to the first output *field* of the previous *IDR view component* in *decoding order* in the same *view* or relative to the first output *field of the previous view component*, in *decoding order* in the same *view*, that contains a *memory management control operation* that marks all *reference pictures* in the *view* as "unused for reference"

3.3.32

prefix NAL unit

NAL unit with `nal_unit_type` equal to 14 that immediately precedes in *decoding order* a *NAL unit* with `nal_unit_type` equal to 1 or 5

Note 1 to entry: The NAL unit that immediately follows in decoding order the prefix NAL unit is referred to as the associated NAL unit. The prefix NAL unit contains data associated with the associated NAL unit, which are considered to be part of the associated NAL unit.

3.3.33

reference picture

view component coded with `nal_ref_idc` greater than 0

Note 1 to entry: A reference picture contains samples that may be used for inter prediction in the decoding process of subsequent view components in decoding order. A reference picture may be an inter-view reference component, in which case the samples contained in the reference picture may also be used for inter-view prediction in the decoding process of subsequent view components in decoding order.

3.3.34

reference picture list

list of *reference pictures* and *inter-view only reference components* that are used for *inter prediction* or *inter-view prediction* of a *P* or *B slice*

Note 1 to entry: For the decoding process of a *P slice*, there is one reference picture list. For the decoding process of a *B slice*, there are two reference picture lists.

3.3.35

reference picture list 0

reference picture list used for *inter prediction* or *inter-view prediction* of a *P* or *B slice*

Note 1 to entry: All inter prediction or inter-view prediction used for *P slices* uses reference picture list 0. Reference picture list 0 is one of two reference picture lists used for inter prediction or inter-view prediction for a *B slice*, with the other being reference picture list 1.

3.3.36

reference picture list 1

reference picture list used for *inter prediction* or *inter-view prediction* of a *B slice*

Note 1 to entry: Reference picture list 1 is one of two reference picture lists used for inter prediction or inter-view prediction for a *B slice*, with the other being reference picture list 0.

3.3.37

reference picture marking

specifies, in the bitstream, how the *decoded view components* are marked for *inter prediction* or *inter-view prediction*

3.3.38

reference processing unit

functional unit that processes an *inter-view prediction reference* before the *inter-view prediction reference* is used for *inter-view prediction* in the *decoding process* of subsequent *view components* in *decoding order*

3.3.39

reference view index

index into a list of *anchor view components* or a list of *non-anchor view components* that are specified in the sequence parameter set MVC extension syntax structure and can be used for *inter-view prediction* as *list 0 prediction* or *list 1 prediction*

3.3.40

right view

right part of a picture coded in a frame-packed manner with the side-by-side frame packing arrangement type or the bottom part of a picture coded in a frame-packed manner with the top-bottom frame packing arrangement type

3.3.41

sub-bitstream

subset of a bitstream

Note 1 to entry: A sub-bitstream is also referred to as a bitstream subset.

3.3.42

subset

subset contains only elements that are also contained in the set from which the subset is derived

Note 1 to entry: The subset may be identical to the set from which it is derived.

3.3.43

subset sequence parameter set

syntax structure containing *syntax elements* that apply to zero or more *non-base views* as determined by the content of a *seq_parameter_set_id syntax element* found in the *picture parameter set* referred to by the *pic_parameter_set_id syntax element* found in each *slice header* of *I, P, and B slices* of a *non-base view component*

3.3.44

target output view

view that is to be output

Note 1 to entry: The target output views are either indicated by external means or, when not indicated by external means, the target output view is the base view.

Note 2 to entry: The output views may be requested by a receiver and may be negotiated between the receiver and the sender.

3.3.45

target temporal level

target temporal level of an *operation point* is the greatest value of *temporal_id* of all *VCL NAL units* in the *bitstream subset* associated with the *operation point*

3.3.46

view

sequence of *view components* associated with an identical value of *view_id*

3.3.47

view component

coded representation of a *view* in a single *access unit*

Note 1 to entry: When *profile_idc* is equal to 134, a view contains samples of two distinct spatially packed constituent frames that are packed into one frame using one of the frame packing arrangement schemes as specified in subclause D.2.26.

3.3.48

view order index

index that indicates the *decoding order* of *view components* in an *access unit*

3.4 Terms related to multiview and depth video coding (Annex H)

3.4.1

depth field view

depth view component of a field

3.4.2

depth frame view

depth view component of a frame

3.4.3

depth view

sequence of *depth view components* associated with an identical value of *view_id*

3.4.4

depth view component

coded representation of the depth of a view in a single access unit

3.4.5

inter-view only reference component

view component, texture view component, or depth view component coded with *nal_ref_idc* equal to 0 and *inter_view_flag* equal to 1

Note 1 to entry: An inter-view only reference component contains samples that may be used for inter-view prediction in the decoding process of subsequent view components in decoding order, but are not used for inter prediction by any view components. Inter-view only reference components are non-reference pictures.

3.4.6

inter-view reference component

view component, texture view component, or depth view component coded with *nal_ref_idc* greater than 0 and *inter_view_flag* equal to 1

Note 1 to entry: An inter-view reference component contains samples that may be used for inter prediction of subsequent pictures in decoding order and inter-view prediction of subsequent view components in decoding order. Inter-view reference components are reference pictures.

3.4.7

MVCD operation point

operation point for which each target output view includes a *texture view* or a *depth view* or both a *texture view* and a *depth view*

3.4.8

MVCD sequence parameter set

collective term for *sequence parameter set* or *subset sequence parameter set*

3.4.9

MVCD sequence parameter set RBSP

collective term for *sequence parameter set RBSP* or *subset sequence parameter set RBSP*

3.4.10

reference picture

view component, texture view component, or depth view component coded with *nal_ref_idc* greater than 0

Note 1 to entry: A reference picture contains samples that may be used for inter prediction in the decoding process of subsequent view components in decoding order. A reference picture may be an inter-view reference component, in which case the samples contained in the reference picture may also be used for inter-view prediction in the decoding process of subsequent view components in decoding order.

3.4.11

stereoscopic texture bitstream

bitstream containing two *texture views*

Note 1 to entry: The bitstream is also conforming to one of the *profiles* specified in Annex G.

3.4.12

texture field view component

texture view component of a field

3.4.13

texture frame view component

texture view component of a frame

3.4.14

texture view

sequence of *texture view components* associated with an identical value of *view_id*

3.4.15

texture view component

coded representation of the texture of a view in a single *access unit*

3.4.16

view

texture view and a *depth view* with the same value of *view_id*, unless explicitly limited to either *texture view* or *depth view*

3.4.17

view component

coded representation of a view in a single *access unit*

Note 1 to entry: A view component may consist of a texture view component and a depth view component.

3.4.18

view component pair

texture view component and a *depth view component* of the same view within the same *access unit*

3.5 Terms related to multiview and depth video with enhanced non-base view coding (Annex I)

3.5.1

MVC texture view component

texture view component composed of *coded slice NAL units* of *nal_unit_type* not equal to 21

3.5.2

texture view component

coded representation of the texture of a view in a single *access unit*

Note 1 to entry: A texture view component may be a 3D-AVC texture view component or an MVC texture view component.

3.5.3

view component pair

texture view component and a *depth view component* of the same view within the same *access unit*

3.5.4

view synthesis prediction

prediction derived from samples of *inter-view reference components* using *motion vectors* derived from a *depth view component* for decoding a *texture view component*

3.5.5

3D-AVC sequence parameter set

collective term for *sequence parameter set* or *subset sequence parameter set*

3.5.6

3D-AVC texture view component

texture view component composed of *coded slice NAL units* of *nal_unit_type* equal to 21

3.5.7

3DV acquisition parameters

closest and farthest depth values

4 Abbreviated terms

CABAC context-based adaptive binary arithmetic coding

CAVLC context-based adaptive variable length coding

CBR	constant bit rate
CPB	coded picture buffer
DPB	decoded picture buffer
DUT	decoder under test
FIFO	first-in, first-out
HRD	hypothetical reference decoder
HSS	hypothetical stream scheduler
IDR	instantaneous decoding refresh
LSB	least significant bit
MB	macroblock
MBAFF	macroblock-adaptive frame-field coding
MFC	multi-resolution frame compatible stereo coding
MSB	most significant bit
MVC	multiview video coding
MVCD	multiview video coding with depth
NAL	network abstraction layer
RBSP	raw byte sequence payload
RPU	reference processing unit
SEI	supplemental enhancement information
SODB	string of data bits
SVC	scalable video coding
UUID	universal unique identifier
VBR	variable bit rate
VCL	video coding layer
VLC	variable length coding
VSP	view synthesis prediction
VSEI	versatile supplemental enhancement information (Rec. ITU-T H.274 ISO/IEC 23002-7)
VUI	video usability information
3D-AVC	multiview video coding with depth as specified in Annex I

5 Conventions

NOTE The mathematical operators used in this document are similar to those used in the C programming language. However, integer division and arithmetic shift operations are specifically defined. Numbering and counting conventions generally begin from 0.

5.1 Arithmetic operators

+	addition
−	subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	multiplication, including matrix multiplication
x^y	exponentiation

Specifies x to the power of y . In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.

$/$ integer division with truncation of the result toward zero
For example, $7/4$ and $-7/4$ are truncated to 1 and $-7/4$ and $7/-4$ are truncated to -1 .

$\div$ division in mathematical formulae where no truncation or rounding is intended.

$\frac{x}{y}$ division in mathematical formulae 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 $y > 0$.

5.2 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

5.3 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.

5.4 Bit-wise operators

$\&$ bit-wise "and"

When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

$|$ bit-wise "or"

When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

$\wedge$ bit-wise "exclusive or"

When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

$x \gg y$ arithmetic right shift of a two's complement integer representation of x by y binary digits

This function is defined only for positive integer values of y . Bits shifted into the 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 positive integer values of y . Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

5.5 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 $x = x - (-3)$

5.6 Range notation

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

5.7 Mathematical functions

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases} \quad (5-1)$$

$\text{Asin}(x)$ the 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. (5-2)

$\text{Atan}(x)$ the 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. (5-3)

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x = 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{otherwise} \end{cases} \quad (5-4)$$

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

$$\text{Clip1}_Y(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_Y) - 1, x) \quad (5-6)$$

$$\text{Clip1}_C(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_C) - 1, x) \quad (5-7)$$

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases} \quad (5-8)$$

$\text{Cos}(x)$ the trigonometric cosine function operating on an argument x in units of radians (5-9)

Floor(x) the greatest integer less than or equal to x. (5-10)

$$\text{InverseRasterScan}(a, b, c, d, e) = \begin{cases} (a \% (d/b)) * b & ; e == 0 \\ (a / (d/b)) * c & ; e == 1 \end{cases} \quad (5-11)$$

Ln(x) returns the natural logarithm of x. (5-12)

Log2(x) returns the base-2 logarithm of x. (5-13)

Log10(x) returns the base-10 logarithm of x. (5-14)

$$\text{Median}(x, y, z) = x + y + z - \text{Min}(x, \text{Min}(y, z)) - \text{Max}(x, \text{Max}(y, z)) \quad (5-15)$$

$$\text{Min}(x, y) = \begin{cases} x & ; x \leq y \\ y & ; x > y \end{cases} \quad (5-16)$$

$$\text{Max}(x, y) = \begin{cases} x & ; x \geq y \\ y & ; x < y \end{cases} \quad (5-17)$$

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5) \quad (5-18)$$

$$\text{Sign}(x) = \begin{cases} 1 & ; x \geq 0 \\ -1 & ; x < 0 \end{cases} \quad (5-19)$$

Sin(x) the trigonometric sine function operating on an argument x in units of radians. (5-20)

Sqrt(x) returns the square root of x. (5-21)

Tan(x) the trigonometric tangent function operating on an argument x in units of radians. (5-22)

5.8 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 5-1 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 5-1 – 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"

5.9 Variables, syntax elements, and tables

Syntax elements in the bitstream are represented in **bold** type. Each syntax element is described by its name (all lower case letters with underscore characters), its one or two syntax categories, and one or two descriptors 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 may use the values of other variables derived from syntax elements values. 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 may 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 may 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 bitstream are referred to as syntax functions. These functions are specified in subclause 7.2 and assume the existence of a bitstream pointer with an indication of the position of the next bit to be read by the decoding process from the bitstream. 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 5.7) 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).

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* may be denoted either as *s*[*x*, *y*] or as *s*_{*yx*}.

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", may be used 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.

5.10 Text description of logical operations

In the text, a statement of logical operations as would be described as follows:

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

may be 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 as follows:

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

may be 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 any 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 as follows:

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

may be described in the following manner:

```
When condition 0, statement 0
When condition 1, statement 1
```

5.11 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 may also have a lower case variable explicitly specified as the 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.

In the specification of a process, a specific macroblock may be referred to by the variable name having a value equal to the address of the specific macroblock.

6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships

6.1 Bitstream formats

This subclause specifies the relationship between the NAL unit stream and byte stream, either of which are referred to as the bitstream.

The bitstream can be in one of two formats: the NAL unit stream format or the byte stream format. The NAL unit stream format is conceptually the more "basic" type. It consists of a sequence of syntax structures called NAL units. This sequence is ordered in decoding order. There are constraints imposed on the decoding order (and contents) of the NAL units in the NAL unit stream.

The byte stream format can be constructed from the NAL unit stream format by ordering the NAL units in decoding order and prefixing each NAL unit with a start code prefix and zero or more zero-valued bytes to form a stream of bytes. The NAL unit stream format can be extracted from the byte stream format by searching for the location of the unique start code prefix pattern within this stream of bytes. Methods of framing the NAL units in a manner other than use of the byte stream format are outside the scope of this document. The byte stream format is specified in Annex B.

6.2 Source, decoded, and output picture formats

This subclause specifies the relationship between source and decoded frames and fields that is given via the bitstream.

The video source that is represented by the bitstream is a sequence of either or both frames or fields (called collectively pictures) in decoding order.

The source and decoded pictures (frames or fields) are each comprised of one or more sample arrays:

- Luma (Y) only (monochrome), with or without an auxiliary array.

- Luma and two Chroma (YCbCr or YCgCo), with or without an auxiliary array.
- Green, Blue and Red (GBR, also known as RGB), with or without an auxiliary array.
- Arrays representing other unspecified monochrome or tri-stimulus colour samplings (for example, YZX, also known as XYZ), with or without an auxiliary array.

For convenience of notation and terminology in this document, the variables and terms associated with these arrays are referred to as luma (or L or Y) and chroma, where the two chroma arrays are referred to as Cb and Cr; regardless of the actual colour representation method in use. The actual colour representation method in use can be indicated in syntax that is specified in Annex E. The (monochrome) auxiliary arrays, which may or may not be present as auxiliary pictures in a coded video sequence, are optional for decoding and can be used for such purposes as alpha blending.

The variables SubWidthC, and SubHeightC are specified in Table 6-1, depending on the chroma format sampling structure, which is specified through chroma_format_idc and separate_colour_plane_flag. An entry marked as "-" in Table 6-1 denotes an undefined value for SubWidthC or SubHeightC. Other values of chroma_format_idc, SubWidthC, and SubHeightC may be specified in the future by ITU-T | ISO/IEC.

Table 6-1 – SubWidthC, and SubHeightC values derived from chroma_format_idc and separate_colour_plane_flag

chroma_format_idc	separate_colour_plane_flag	Chroma Format	SubWidthC	SubHeightC
0	0	monochrome	-	-
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	-	-

In monochrome sampling there is only one sample array, which is nominally considered the luma array.

In 4:2:0 sampling, each of the two chroma arrays has half the height and half the width of the luma array.

In 4:2:2 sampling, each of the two chroma arrays has the same height and half the width of the luma array.

In 4:4:4 sampling, depending on the value of separate_colour_plane_flag, the following applies:

- If separate_colour_plane_flag is equal to 0, each of the two chroma arrays has the same height and width as the luma array.
- Otherwise (separate_colour_plane_flag is equal to 1), the three colour planes are separately processed as monochrome sampled pictures.

The width and height of the luma sample arrays are each an integer multiple of 16. In coded video sequences using 4:2:0 chroma sampling, the width and height of chroma sample arrays are each an integer multiple of 8. In coded video sequences using 4:2:2 sampling, the width of the chroma sample arrays is an integer multiple of 8 and the height is an integer multiple of 16. The height of a luma array that is coded as two separate fields or in macroblock-adaptive frame-field coding (see below) is an integer multiple of 32. In coded video sequences using 4:2:0 chroma sampling, the height of each chroma array that is coded as two separate fields or in macroblock-adaptive frame-field coding (see below) is an integer multiple of 16. The width or height of pictures output from the decoding process need not be an integer multiple of 16 and can be specified using a cropping rectangle.

The syntax for the luma and (when present) chroma arrays are ordered such when data for all three colour components is present, the data for the luma array is first, followed by any data for the Cb array, followed by any data for the Cr array, unless otherwise specified.

The width of fields coded referring to a specific sequence parameter set is the same as that of frames coded referring to the same sequence parameter set (see below). The height of fields coded referring to a specific sequence parameter set is half that of frames coded referring to the same sequence parameter set (see below).

The number of bits necessary for the representation of each of the samples in the luma and chroma arrays in a video sequence is in the range of 8 to 14, and the number of bits used in the luma array may differ from the number of bits used in the chroma arrays.