

Australian Standard™

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

**Part 10: Advanced Video Coding**

**STANDARDS**  
Australia



This Australian Standard was prepared by Committee IT-029, Coded Representation of Picture, Audio and Multimedia/Hypermedia Information. It was approved on behalf of the Council of Standards Australia on 12 May 2006.  
This Standard was published on 2 June 2006.

---

The following are represented on Committee IT-029:

Australian Broadcasting Authority (ABA)  
Australian Broadcasting Corporation (ABC)  
Australian Consumers Association  
Australian Subscription Television  
CSIRO Mathematical & Information Services  
Department of Defence (Australia)  
Free TV Australia  
School of Computer Science and Mathematics  
Victoria University of Technology  
Special Broadcasting Service (SBS)  
The University of New South Wales  
University of Wollongong

---

### **Keeping Standards up-to-date**

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about Standards can be found by visiting the Standards Web Shop at [www.standards.com.au](http://www.standards.com.au) and looking up the relevant Standard in the on-line catalogue.

Alternatively, the printed Catalogue provides information current at 1 January each year, and the monthly magazine, *The Global Standard*, has a full listing of revisions and amendments published each month.

Australian Standards™ and other products and services developed by Standards Australia are published and distributed under contract by SAI Global, which operates the Standards Web Shop.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Contact us via email at [mail@standards.org.au](mailto:mail@standards.org.au), or write to the Chief Executive, Standards Australia, GPO Box 476, Sydney, NSW 2001.

---

*This Standard was issued in draft form for comment as DR 06118.*

Australian Standard™

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

**Part 10: Advanced Video Coding**

First published as AS ISO/IEC 14496.10—2006.

**COPYRIGHT**

© Standards Australia

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Published by Standards Australia GPO Box 476, Sydney, NSW 2001, Australia

ISBN 0 7337 7471 7

## PREFACE

This Standard was prepared by the Standards Australia Committee IT-029, Coded Representation of Picture, Audio and Multimedia/Hypermedia Information.

This Standard is identical with, and has been reproduced from ISO/IEC 14496-10:2005, *Information technology—Coding of audio-visual objects, Part 10: Advanced Video Coding*.

The objective of this Standard is to provide the multimedia community with the mechanism of 'Advanced Video Coding', which facilitates video data interchange among different applications, including services using video over DSL networks and handheld devices with the ultimate goal of providing bandwidth cost savings for the community.

Standards Australia is in the process of adopting most of ISO/IEC 14496 series Standards. Refer to the website for current list.

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover
- (b) In the source text 'this part of ISO/IEC 14499' should read 'this Australian Standard'.
- (c) A full point substitutes for a comma when referring to a decimal marker.

None of the normative references in the source document have been adopted as Australian or Australian/New Zealand Standards.

## CONTENTS

	<i>Page</i>
<b>0 Introduction.....</b>	<b>xiv</b>
0.1 Prologue.....	xiv
0.2 Purpose.....	xiv
0.3 Applications.....	xiv
0.4 Publication and versions of this International Standard.....	xiv
0.5 Profiles and levels.....	xv
0.6 Overview of the design characteristics.....	xv
0.6.1 Predictive coding.....	xv
0.6.2 Coding of progressive and interlaced video.....	xvi
0.6.3 Picture partitioning into macroblocks and smaller partitions.....	xv
0.6.4 Spatial redundancy reduction.....	xvi
0.7 How to read this specification.....	xvi
<b>1 Scope.....</b>	<b>1</b>
<b>2 Normative references.....</b>	<b>1</b>
<b>3 Definitions.....</b>	<b>1</b>
<b>4 Abbreviations.....</b>	<b>9</b>
<b>5 Conventions.....</b>	<b>10</b>
5.1 Arithmetic operators.....	10
5.2 Logical operators.....	10
5.3 Relational operators.....	11
5.4 Bit-wise operators.....	11
5.5 Assignment operators.....	11
5.6 Range notation.....	11
5.7 Mathematical functions.....	11
5.8 Variables, syntax elements, and tables.....	12
5.9 Text description of logical operations.....	13
5.10 Processes.....	14
<b>6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships.....</b>	<b>14</b>
6.1 Bitstream formats.....	14
6.2 Source, decoded, and output picture formats.....	15
6.3 Spatial subdivision of pictures and slices.....	20
6.4 Inverse scanning processes and derivation processes for neighbours.....	20
6.4.1 Inverse macroblock scanning processes.....	21
6.4.2 Inverse macroblock partition and sub-macroblock partition scanning process.....	21
6.4.2.1 Inverse macroblock partition scanning process.....	22
6.4.2.2 Inverse sub-macroblock partition scanning process.....	22
6.4.3 Inverse 4x4 luma block scanning process.....	23
6.4.4 Inverse 8x8 luma block scanning process.....	23
6.4.5 Derivation process of the availability for macroblock addresses.....	23
6.4.6 Derivation process for neighbouring macroblock addresses and their availability.....	24
6.4.7 Derivation process for neighbouring macroblock addresses and their availability in MBAFF frames.....	24
6.4.8 Derivation processes for neighbouring macroblocks, blocks, and partitions.....	25
6.4.8.1 Derivation process for neighbouring macroblocks.....	26
6.4.8.2 Derivation process for neighbouring 8x8 luma block.....	26
6.4.8.3 Derivation process for neighbouring 4x4 luma blocks.....	27
6.4.8.4 Derivation process for neighbouring 4x4 chroma blocks.....	28
6.4.8.5 Derivation process for neighbouring partitions.....	28
6.4.9 Derivation process for neighbouring locations.....	30
6.4.9.1 Specification for neighbouring locations in fields and non-MBAFF frames.....	30
6.4.9.2 Specification for neighbouring locations in MBAFF frames.....	31

<b>7</b>	<b>Syntax and semantics</b>	<b>33</b>
7.1	Method of describing syntax in tabular form	33
7.2	Specification of syntax functions, categories, and descriptors	34
7.3	Syntax in tabular form	36
7.3.1	NAL unit syntax	36
7.3.2	Raw byte sequence payloads and RBSP trailing bits syntax	37
7.3.2.1	Sequence parameter set RBSP syntax	37
7.3.2.1.1	Scaling list syntax	38
7.3.2.1.2	Sequence parameter set extension RBSP syntax	39
7.3.2.2	Picture parameter set RBSP syntax	39
7.3.2.3	Supplemental enhancement information RBSP syntax	40
7.3.2.3.1	Supplemental enhancement information message syntax	41
7.3.2.4	Access unit delimiter RBSP syntax	41
7.3.2.5	End of sequence RBSP syntax	41
7.3.2.6	End of stream RBSP syntax	41
7.3.2.7	Filler data RBSP syntax	42
7.3.2.8	Slice layer without partitioning RBSP syntax	42
7.3.2.9	Slice data partition RBSP syntax	42
7.3.2.9.1	Slice data partition A RBSP syntax	42
7.3.2.9.2	Slice data partition B RBSP syntax	42
7.3.2.9.3	Slice data partition C RBSP syntax	43
7.3.2.10	RBSP slice trailing bits syntax	43
7.3.2.11	RBSP trailing bits syntax	43
7.3.3	Slice header syntax	44
7.3.3.1	Reference picture list reordering syntax	45
7.3.3.2	Prediction weight table syntax	46
7.3.3.3	Decoded reference picture marking syntax	47
7.3.4	Slice data syntax	48
7.3.5	Macroblock layer syntax	49
7.3.5.1	Macroblock prediction syntax	50
7.3.5.2	Sub-macroblock prediction syntax	51
7.3.5.3	Residual data syntax	52
7.3.5.3.1	Residual block CAVLC syntax	53
7.3.5.3.2	Residual block CABAC syntax	55
7.4	Semantics	56
7.4.1	NAL unit semantics	56
7.4.1.1	Encapsulation of an SODB within an RBSP (informative)	58
7.4.1.2	Order of NAL units and association to coded pictures, access units, and video sequences	59
7.4.1.2.1	Order of sequence and picture parameter set RBSPs and their activation	59
7.4.1.2.2	Order of access units and association to coded video sequences	60
7.4.1.2.3	Order of NAL units and coded pictures and association to access units	61
7.4.1.2.4	Detection of the first VCL NAL unit of a primary coded picture	62
7.4.1.2.5	Order of VCL NAL units and association to coded pictures	63
7.4.2	Raw byte sequence payloads and RBSP trailing bits semantics	63
7.4.2.1	Sequence parameter set RBSP semantics	63
7.4.2.1.1	Scaling list semantics	68
7.4.2.1.2	Sequence parameter set extension RBSP semantics	69
7.4.2.2	Picture parameter set RBSP semantics	70
7.4.2.3	Supplemental enhancement information RBSP semantics	73
7.4.2.3.1	Supplemental enhancement information message semantics	73
7.4.2.4	Access unit delimiter RBSP semantics	73
7.4.2.5	End of sequence RBSP semantics	73
7.4.2.6	End of stream RBSP semantics	74
7.4.2.7	Filler data RBSP semantics	74
7.4.2.8	Slice layer without partitioning RBSP semantics	74
7.4.2.9	Slice data partition RBSP semantics	74
7.4.2.9.1	Slice data partition A RBSP semantics	74
7.4.2.9.2	Slice data partition B RBSP semantics	74
7.4.2.9.3	Slice data partition C RBSP semantics	74

7.4.2.10	RBSP slice trailing bits semantics.....	75
7.4.2.11	RBSP trailing bits semantics.....	75
7.4.3	Slice header semantics.....	75
7.4.3.1	Reference picture list reordering semantics.....	81
7.4.3.2	Prediction weight table semantics.....	82
7.4.3.3	Decoded reference picture marking semantics.....	82
7.4.4	Slice data semantics.....	85
7.4.5	Macroblock layer semantics.....	86
7.4.5.1	Macroblock prediction semantics.....	93
7.4.5.2	Sub-macroblock prediction semantics.....	94
7.4.5.3	Residual data semantics.....	96
7.4.5.3.1	Residual block CAVLC semantics.....	97
7.4.5.3.2	Residual block CABAC semantics.....	97
<b>8</b>	<b>Decoding process.....</b>	<b>98</b>
8.1	NAL unit decoding process.....	99
8.2	Slice decoding process.....	99
8.2.1	Decoding process for picture order count.....	99
8.2.1.1	Decoding process for picture order count type 0.....	101
8.2.1.2	Decoding process for picture order count type 1.....	102
8.2.1.3	Decoding process for picture order count type 2.....	103
8.2.2	Decoding process for macroblock to slice group map.....	104
8.2.2.1	Specification for interleaved slice group map type.....	105
8.2.2.2	Specification for dispersed slice group map type.....	105
8.2.2.3	Specification for foreground with left-over slice group map type.....	105
8.2.2.4	Specification for box-out slice group map types.....	106
8.2.2.5	Specification for raster scan slice group map types.....	106
8.2.2.6	Specification for wipe slice group map types.....	106
8.2.2.7	Specification for explicit slice group map type.....	107
8.2.2.8	Specification for conversion of map unit to slice group map to macroblock to slice group map.....	107
8.2.3	Decoding process for slice data partitioning.....	107
8.2.4	Decoding process for reference picture lists construction.....	108
8.2.4.1	Decoding process for picture numbers.....	108
8.2.4.2	Initialisation process for reference picture lists.....	109
8.2.4.2.1	Initialisation process for the reference picture list for P and SP slices in frames.....	109
8.2.4.2.2	Initialisation process for the reference picture list for P and SP slices in fields.....	110
8.2.4.2.3	Initialisation process for reference picture lists for B slices in frames.....	110
8.2.4.2.4	Initialisation process for reference picture lists for B slices in fields.....	111
8.2.4.2.5	Initialisation process for reference picture lists in fields.....	112
8.2.4.3	Reordering process for reference picture lists.....	113
8.2.4.3.1	Reordering process of reference picture lists for short-term reference pictures.....	113
8.2.4.3.2	Reordering process of reference picture lists for long-term reference pictures.....	114
8.2.5	Decoded reference picture marking process.....	114
8.2.5.1	Sequence of operations for decoded reference picture marking process.....	115
8.2.5.2	Decoding process for gaps in frame_num.....	115
8.2.5.3	Sliding window decoded reference picture marking process.....	116
8.2.5.4	Adaptive memory control decoded reference picture marking process.....	116
8.2.5.4.1	Marking process of a short-term reference picture as “unused for reference”.....	116
8.2.5.4.2	Marking process of a long-term reference picture as “unused for reference”.....	117
8.2.5.4.3	Assignment process of a LongTermFrameIdx to a short-term reference picture.....	117
8.2.5.4.4	Decoding process for MaxLongTermFrameIdx.....	117
8.2.5.4.5	Marking process of all reference pictures as “unused for reference” and setting MaxLongTermFrameIdx to “no long-term frame indices”.....	118
8.2.5.4.6	Process for assigning a long-term frame index to the current picture.....	118
8.3	Intra prediction process.....	118
8.3.1	Intra_4x4 prediction process for luma samples.....	119
8.3.1.1	Derivation process for the Intra4x4PredMode.....	119
8.3.1.2	Intra_4x4 sample prediction.....	121
8.3.1.2.1	Specification of Intra_4x4_Vertical prediction mode.....	122
8.3.1.2.2	Specification of Intra_4x4_Horizontal prediction mode.....	122
8.3.1.2.3	Specification of Intra_4x4_DC prediction mode.....	122

8.3.1.2.4	Specification of Intra_4x4_Diagonal_Down_Left prediction mode .....	123
8.3.1.2.5	Specification of Intra_4x4_Diagonal_Down_Right prediction mode .....	123
8.3.1.2.6	Specification of Intra_4x4_Vertical_Right prediction mode .....	123
8.3.1.2.7	Specification of Intra_4x4_Horizontal_Down prediction mode .....	124
8.3.1.2.8	Specification of Intra_4x4_Vertical_Left prediction mode .....	124
8.3.1.2.9	Specification of Intra_4x4_Horizontal_Up prediction mode .....	124
8.3.2	Intra_8x8 prediction process for luma samples .....	125
8.3.2.1	Derivation process for Intra8x8PredMode .....	125
8.3.2.2	Intra_8x8 sample prediction .....	127
8.3.2.2.1	Reference sample filtering process for Intra_8x8 sample prediction .....	128
8.3.2.2.2	Specification of Intra_8x8_Vertical prediction mode .....	129
8.3.2.2.3	Specification of Intra_8x8_Horizontal prediction mode .....	129
8.3.2.2.4	Specification of Intra_8x8_DC prediction mode .....	129
8.3.2.2.5	Specification of Intra_8x8_Diagonal_Down_Left prediction mode .....	130
8.3.2.2.6	Specification of Intra_8x8_Diagonal_Down_Right prediction mode .....	130
8.3.2.2.7	Specification of Intra_8x8_Vertical_Right prediction mode .....	130
8.3.2.2.8	Specification of Intra_8x8_Horizontal_Down prediction mode .....	131
8.3.2.2.9	Specification of Intra_8x8_Vertical_Left prediction mode .....	131
8.3.2.2.10	Specification of Intra_8x8_Horizontal_Up prediction mode .....	131
8.3.3	Intra_16x16 prediction process for luma samples .....	132
8.3.3.1	Specification of Intra_16x16_Vertical prediction mode .....	133
8.3.3.2	Specification of Intra_16x16_Horizontal prediction mode .....	133
8.3.3.3	Specification of Intra_16x16_DC prediction mode .....	133
8.3.3.4	Specification of Intra_16x16_Plane prediction mode .....	133
8.3.4	Intra prediction process for chroma samples .....	134
8.3.4.1	Specification of Intra_Chroma_DC prediction mode .....	134
8.3.4.2	Specification of Intra_Chroma_Horizontal prediction mode .....	136
8.3.4.3	Specification of Intra_Chroma_Vertical prediction mode .....	136
8.3.4.4	Specification of Intra_Chroma_Plane prediction mode .....	136
8.3.5	Sample construction process for I_PCM macroblocks .....	137
8.4	Inter prediction process .....	137
8.4.1	Derivation process for motion vector components and reference indices .....	139
8.4.1.1	Derivation process for luma motion vectors for skipped macroblocks in P and SP slices .....	140
8.4.1.2	Derivation process for luma motion vectors for B_Skip, B_Direct_16x16, and B_Direct_8x8 .....	141
8.4.1.2.1	Derivation process for the co-located 4x4 sub-macroblock partitions .....	141
8.4.1.2.2	Derivation process for spatial direct luma motion vector and reference index prediction mode .....	144
8.4.1.2.3	Derivation process for temporal direct luma motion vector and reference index prediction mode .....	146
8.4.1.3	Derivation process for luma motion vector prediction .....	148
8.4.1.3.1	Derivation process for median luma motion vector prediction .....	149
8.4.1.3.2	Derivation process for motion data of neighbouring partitions .....	150
8.4.1.4	Derivation process for chroma motion vectors .....	151
8.4.2	Decoding process for inter prediction samples .....	151
8.4.2.1	Reference picture selection process .....	152
8.4.2.2	Fractional sample interpolation process .....	153
8.4.2.2.1	Luma sample interpolation process .....	154
8.4.2.2.2	Chroma sample interpolation process .....	157
8.4.2.3	Weighted sample prediction process .....	158
8.4.2.3.1	Default weighted sample prediction process .....	158
8.4.2.3.2	Weighted sample prediction process .....	159
8.5	Transform coefficient decoding process and picture construction process prior to deblocking filter process ..	161
8.5.1	Specification of transform decoding process for 4x4 luma residual blocks .....	162
8.5.2	Specification of transform decoding process for luma samples of Intra_16x16 macroblock prediction mode .....	162
8.5.3	Specification of transform decoding process for 8x8 luma residual blocks .....	163
8.5.4	Specification of transform decoding process for chroma samples .....	164
8.5.5	Inverse scanning process for transform coefficients .....	166
8.5.6	Inverse scanning process for 8x8 luma transform coefficients .....	167
8.5.7	Derivation process for the chroma quantisation parameters and scaling function .....	169

	Page	
8.5.8	Scaling and transformation process for luma DC transform coefficients for Intra_16x16 macroblock type .....	171
8.5.9	Scaling and transformation process for chroma DC transform coefficients.....	171
8.5.10	Scaling and transformation process for residual 4x4 blocks.....	173
8.5.11	Scaling and transformation process for residual 8x8 luma blocks .....	175
8.5.12	Picture construction process prior to deblocking filter process.....	178
8.5.13	Residual colour transform process .....	179
8.6	Decoding process for P macroblocks in SP slices or SI macroblocks.....	179
8.6.1	SP decoding process for non-switching pictures.....	180
8.6.1.1	Luma transform coefficient decoding process .....	180
8.6.1.2	Chroma transform coefficient decoding process.....	181
8.6.2	SP and SI slice decoding process for switching pictures .....	182
8.6.2.1	Luma transform coefficient decoding process .....	182
8.6.2.2	Chroma transform coefficient decoding process.....	183
8.7	Deblocking filter process .....	184
8.7.1	Filtering process for block edges .....	188
8.7.2	Filtering process for a set of samples across a horizontal or vertical block edge.....	189
8.7.2.1	Derivation process for the luma content dependent boundary filtering strength.....	190
8.7.2.2	Derivation process for the thresholds for each block edge.....	191
8.7.2.3	Filtering process for edges with bS less than 4 .....	193
8.7.2.4	Filtering process for edges for bS equal to 4.....	194
<b>9</b>	<b>Parsing process.....</b>	<b>195</b>
9.1	Parsing process for Exp-Golomb codes .....	195
9.1.1	Mapping process for signed Exp-Golomb codes .....	197
9.1.2	Mapping process for coded block pattern .....	197
9.2	CAVLC parsing process for transform coefficient levels .....	200
9.2.1	Parsing process for total number of transform coefficient levels and trailing ones .....	201
9.2.2	Parsing process for level information .....	205
9.2.2.1	Parsing process for level_prefix.....	206
9.2.3	Parsing process for run information .....	206
9.2.4	Combining level and run information.....	209
9.3	CABAC parsing process for slice data.....	209
9.3.1	Initialisation process .....	210
9.3.1.1	Initialisation process for context variables.....	211
9.3.1.2	Initialisation process for the arithmetic decoding engine.....	222
9.3.2	Binarization process.....	222
9.3.2.1	Unary (U) binarization process.....	224
9.3.2.2	Truncated unary (TU) binarization process.....	224
9.3.2.3	Concatenated unary/truncated unary Exp-Golomb (UEGk) binarization process.....	225
9.3.2.4	Fixed-length (FL) binarization process.....	225
9.3.2.5	Binarization process for macroblock type and sub-macroblock type.....	226
9.3.2.6	Binarization process for coded block pattern.....	229
9.3.2.7	Binarization process for mb_qp_delta.....	229
9.3.3	Decoding process flow.....	230
9.3.3.1	Derivation process for ctxIdx.....	230
9.3.3.1.1	Assignment process of ctxIdxInc using neighbouring syntax elements .....	232
9.3.3.1.1.1	Derivation process of ctxIdxInc for the syntax element mb_skip_flag.....	232
9.3.3.1.1.2	Derivation process of ctxIdxInc for the syntax element mb_field_decoding_flag.....	232
9.3.3.1.1.3	Derivation process of ctxIdxInc for the syntax element mb_type .....	233
9.3.3.1.1.4	Derivation process of ctxIdxInc for the syntax element coded_block_pattern.....	233
9.3.3.1.1.5	Derivation process of ctxIdxInc for the syntax element mb_qp_delta .....	234
9.3.3.1.1.6	Derivation process of ctxIdxInc for the syntax elements ref_idx_l0 and ref_idx_l1.....	234
9.3.3.1.1.7	Derivation process of ctxIdxInc for the syntax elements mvd_l0 and mvd_l1 .....	235
9.3.3.1.1.8	Derivation process of ctxIdxInc for the syntax element intra_chroma_pred_mode.....	237
9.3.3.1.1.9	Derivation process of ctxIdxInc for the syntax element coded_block_flag .....	237
9.3.3.1.1.10	Derivation process of ctxIdxInc for the syntax element transform_size_8x8_flag .....	238
9.3.3.1.2	Assignment process of ctxIdxInc using prior decoded bin values.....	239
9.3.3.1.3	Assignment process of ctxIdxInc for syntax elements significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1 .....	239
9.3.3.2	Arithmetic decoding process.....	242

9.3.3.2.1	Arithmetic decoding process for a binary decision .....	243
9.3.3.2.1.1	State transition process .....	243
9.3.3.2.2	Renormalization process in the arithmetic decoding engine .....	246
9.3.3.2.3	Bypass decoding process for binary decisions .....	247
9.3.3.2.4	Decoding process for binary decisions before termination .....	247
9.3.4	Arithmetic encoding process (informative) .....	248
9.3.4.1	Initialisation process for the arithmetic encoding engine (informative) .....	248
9.3.4.2	Encoding process for a binary decision (informative) .....	248
9.3.4.3	Renormalization process in the arithmetic encoding engine (informative) .....	249
9.3.4.4	Bypass encoding process for binary decisions (informative) .....	251
9.3.4.5	Encoding process for a binary decision before termination (informative) .....	252
9.3.4.6	Byte stuffing process (informative) .....	253
<b>Annex A</b>	<b>Profiles and levels .....</b>	<b>255</b>
A.1	Requirements on video decoder capability .....	255
A.2	Profiles .....	255
A.2.1	Baseline profile .....	255
A.2.2	Main profile .....	256
A.2.3	Extended profile .....	256
A.2.4	High profile .....	256
A.2.5	High 10 profile .....	257
A.2.6	High 4:2:2 profile .....	257
A.2.7	High 4:4:4 profile .....	258
A.3	Levels .....	258
A.3.1	Level limits common to the Baseline, Main, and Extended profiles .....	258
A.3.2	Level limits common to the High, High 10, High 4:2:2, and High 4:4:4 profiles .....	260
A.3.3	Profile-specific level limits .....	261
A.3.3.1	Baseline profile limits .....	262
A.3.3.2	Main, High, High 10, High 4:2:2, or High 4:4:4 profile limits .....	263
A.3.3.3	Extended Profile Limits .....	264
A.3.4	Effect of level limits on frame rate (informative) .....	266
<b>Annex B</b>	<b>Byte stream format .....</b>	<b>269</b>
B.1	Byte stream NAL unit syntax and semantics .....	269
B.1.1	Byte stream NAL unit syntax .....	269
B.1.2	Byte stream NAL unit semantics .....	269
B.2	Byte stream NAL unit decoding process .....	270
B.3	Decoder byte-alignment recovery (informative) .....	270
<b>Annex C</b>	<b>Hypothetical reference decoder .....</b>	<b>272</b>
C.1	Operation of coded picture buffer (CPB) .....	274
C.1.1	Timing of bitstream arrival .....	274
C.1.2	Timing of coded picture removal .....	275
C.2	Operation of the decoded picture buffer (DPB) .....	276
C.2.1	Decoding of gaps in frame_num and storage of "non-existing" frames .....	276
C.2.2	Picture decoding and output .....	276
C.2.3	Removal of pictures from the DPB before possible insertion of the current picture .....	277
C.2.4	Current decoded picture marking and storage .....	277
C.2.4.1	Marking and storage of a reference decoded picture into the DPB .....	277
C.2.4.2	Storage of a non-reference picture into the DPB .....	277
C.3	Bitstream conformance .....	277
C.4	Decoder conformance .....	279
C.4.1	Operation of the output order DPB .....	280
C.4.2	Decoding of gaps in frame_num and storage of "non-existing" pictures .....	280
C.4.3	Picture decoding .....	280
C.4.4	Removal of pictures from the DPB before possible insertion of the current picture .....	280
C.4.5	Current decoded picture marking and storage .....	281
C.4.5.1	Storage and marking of a reference decoded picture into the DPB .....	281
C.4.5.2	Storage and marking of a non-reference decoded picture into the DPB .....	281
C.4.5.3	"Bumping" process .....	281

<b>Annex D Supplemental enhancement information .....</b>	<b>283</b>
D.1 SEI payload syntax .....	284
D.1.1 Buffering period SEI message syntax .....	285
D.1.2 Picture timing SEI message syntax .....	285
D.1.3 Pan-scan rectangle SEI message syntax .....	286
D.1.4 Filler payload SEI message syntax .....	287
D.1.5 User data registered by ITU-T Recommendation T.35 SEI message syntax .....	287
D.1.6 User data unregistered SEI message syntax .....	287
D.1.7 Recovery point SEI message syntax .....	287
D.1.8 Decoded reference picture marking repetition SEI message syntax .....	288
D.1.9 Spare picture SEI message syntax .....	288
D.1.10 Scene information SEI message syntax .....	289
D.1.11 Sub-sequence information SEI message syntax .....	289
D.1.12 Sub-sequence layer characteristics SEI message syntax .....	289
D.1.13 Sub-sequence characteristics SEI message syntax .....	290
D.1.14 Full-frame freeze SEI message syntax .....	290
D.1.15 Full-frame freeze release SEI message syntax .....	290
D.1.16 Full-frame snapshot SEI message syntax .....	290
D.1.17 Progressive refinement segment start SEI message syntax .....	291
D.1.18 Progressive refinement segment end SEI message syntax .....	291
D.1.19 Motion-constrained slice group set SEI message syntax .....	291
D.1.20 Film grain characteristics SEI message syntax .....	292
D.1.21 Deblocking filter display preference SEI message syntax .....	292
D.1.22 Stereo video information SEI message syntax .....	293
D.1.23 Reserved SEI message syntax .....	293
D.2 SEI payload semantics .....	293
D.2.1 Buffering period SEI message semantics .....	293
D.2.2 Picture timing SEI message semantics .....	294
D.2.3 Pan-scan rectangle SEI message semantics .....	297
D.2.4 Filler payload SEI message semantics .....	299
D.2.5 User data registered by ITU-T Recommendation T.35 SEI message semantics .....	299
D.2.6 User data unregistered SEI message semantics .....	299
D.2.7 Recovery point SEI message semantics .....	299
D.2.8 Decoded reference picture marking repetition SEI message semantics .....	301
D.2.9 Spare picture SEI message semantics .....	301
D.2.10 Scene information SEI message semantics .....	303
D.2.11 Sub-sequence information SEI message semantics .....	304
D.2.12 Sub-sequence layer characteristics SEI message semantics .....	306
D.2.13 Sub-sequence characteristics SEI message semantics .....	307
D.2.14 Full-frame freeze SEI message semantics .....	308
D.2.15 Full-frame freeze release SEI message semantics .....	308
D.2.16 Full-frame snapshot SEI message semantics .....	309
D.2.17 Progressive refinement segment start SEI message semantics .....	309
D.2.18 Progressive refinement segment end SEI message semantics .....	309
D.2.19 Motion-constrained slice group set SEI message semantics .....	310
D.2.20 Film grain characteristics SEI message semantics .....	310
D.2.21 Deblocking filter display preference SEI message semantics .....	316
D.2.22 Stereo video information SEI message semantics .....	318
D.2.23 Reserved SEI message semantics .....	318
<b>Annex E Video usability information .....</b>	<b>319</b>
E.1 VUI syntax .....	320
E.1.1 VUI parameters syntax .....	320
E.1.2 HRD parameters syntax .....	321
E.2 VUI semantics .....	321
E.2.1 VUI parameters semantics .....	321
E.2.2 HRD parameters semantics .....	332
<b>Annex F Patent Rights .....</b>	<b>334</b>

**LIST OF FIGURES**

*Page*

Figure 6-1 – Nominal vertical and horizontal locations of 4:2:0 luma and chroma samples in a frame .....	16
Figure 6-2 – Nominal vertical and horizontal sampling locations of 4:2:0 samples in top and bottom fields. ....	17
Figure 6-3 – Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a frame .....	18
Figure 6-4 – Nominal vertical and horizontal sampling locations of 4:2:2 samples top and bottom fields .....	18
Figure 6-5 – Nominal vertical and horizontal locations of 4:4:4 luma and chroma samples in a frame .....	19
Figure 6-6 – Nominal vertical and horizontal sampling locations of 4:4:4 samples top and bottom fields .....	19
Figure 6-7 – A picture with 11 by 9 macroblocks that is partitioned into two slices .....	20
Figure 6-8 – Partitioning of the decoded frame into macroblock pairs .....	20
Figure 6-9 – Macroblock partitions, sub-macroblock partitions, macroblock partition scans, and sub-macroblock partition scans .....	22
Figure 6-10 – Scan for 4x4 luma blocks .....	23
Figure 6-11 – Scan for 8x8 luma blocks .....	23
Figure 6-12 – Neighbouring macroblocks for a given macroblock.....	24
Figure 6-13 – Neighbouring macroblocks for a given macroblock in MBAFF frames .....	25
Figure 6-14 – Determination of the neighbouring macroblock, blocks, and partitions (informative).....	26
Figure 7-1 – Structure of an access unit not containing any NAL units with nal_unit_type equal to 0, 7, 8, or in the range of 12 to 18, inclusive, or in the range of 20 to 31, inclusive. ....	62
Figure 8-1 – Intra_4x4 prediction mode directions (informative).....	120
Figure 8-2 –Example for temporal direct-mode motion vector inference (informative).....	148
Figure 8-3 – Directional segmentation prediction (informative).....	149
Figure 8-4 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation.....	155
Figure 8-5 – Fractional sample position dependent variables in chroma interpolation and surrounding integer position samples A, B, C, and D .....	157
Figure 8-6 – Assignment of the indices of dcY to luma4x4BlkIdx.....	163
Figure 8-7 – Assignment of the indices of dcC to chroma4x4BlkIdx: (a) chroma_format_idc equal to 1, (b) chroma_format_idc equal to 2, (c) chroma_format_idc equal to 3 .....	165
Figure 8-8 – 4x4 block scans. (a) Zig-zag scan. (b) Field scan (informative).....	166
Figure 8-9 – 8x8 block scans. (a) 8x8 zig-zag scan. (b) 8x8 field scan (informative) .....	167
Figure 8-10 – Boundaries in a macroblock to be filtered .....	185
Figure 8-11 – Convention for describing samples across a 4x4 block horizontal or vertical boundary.....	189
Figure 9-1 – Illustration of CABAC parsing process for a syntax element SE (informative).....	210
Figure 9-2 – Overview of the arithmetic decoding process for a single bin (informative) .....	242
Figure 9-3 – Flowchart for decoding a decision.....	244
Figure 9-4 – Flowchart of renormalization .....	246
Figure 9-5 – Flowchart of bypass decoding process .....	247
Figure 9-6 – Flowchart of decoding a decision before termination .....	248
Figure 9-7 – Flowchart for encoding a decision.....	249

	<i>Page</i>
Figure 9-8 – Flowchart of renormalization in the encoder.....	250
Figure 9-9 – Flowchart of PutBit(B).....	251
Figure 9-10 – Flowchart of encoding bypass.....	252
Figure 9-11 – Flowchart of encoding a decision before termination.....	253
Figure 9-12 – Flowchart of flushing at termination.....	253
Figure C-1 – Structure of byte streams and NAL unit streams for HRD conformance checks.....	272
Figure C-2 – HRD buffer model.....	273
Figure E-1 – Location of chroma samples for top and bottom fields as a function of chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field.....	328

### LIST OF TABLES

Table 6-1 –SubWidthC, and SubHeightC values derived from chroma_format_idc.....	15
Table 6-2 – Specification of input and output assignments for subclauses 6.4.8.1 to 6.4.8.5.....	26
Table 6-3 – Specification of mbAddrN.....	30
Table 6-4 - Specification of mbAddrN and yM.....	32
Table 7-1 – NAL unit type codes.....	57
Table 7-2 – Assignment of mnemonic names to scaling list indices and specification of fall-back rule.....	65
Table 7-3 – Specification of default scaling lists Default_4x4_Intra and Default_4x4_Inter.....	66
Table 7-4 – Specification of default scaling lists Default_8x8_Intra and Default_8x8_Inter.....	66
Table 7-5 – Meaning of primary_pic_type.....	73
Table 7-6 – Name association to slice_type.....	76
Table 7-7 – Reordering_of_pic_nums_idc operations for reordering of reference picture lists.....	81
Table 7-8 – Interpretation of adaptive_ref_pic_marking_mode_flag.....	83
Table 7-9 – Memory management control operation (memory_management_control_operation) values.....	84
Table 7-10 – Allowed collective macroblock types for slice_type.....	86
Table 7-11 – Macroblock types for I slices.....	88
Table 7-12 – Macroblock type with value 0 for SI slices.....	89
Table 7-13 – Macroblock type values 0 to 4 for P and SP slices.....	90
Table 7-14 – Macroblock type values 0 to 22 for B slices.....	91
Table 7-15 – Specification of CodedBlockPatternChroma values.....	93
Table 7-16 – Relationship between intra_chroma_pred_mode and spatial prediction modes.....	93
Table 7-17 – Sub-macroblock types in P macroblocks.....	94
Table 7-18 – Sub-macroblock types in B macroblocks.....	95
Table 8-1 – Refined slice group map type.....	104
Table 8-2 – Specification of Intra4x4PredMode[ luma4x4BlkIdx ] and associated names.....	120
Table 8-3 – Specification of Intra8x8PredMode[ luma8x8BlkIdx ] and associated names.....	126
Table 8-4 – Specification of Intra16x16PredMode and associated names.....	132
Table 8-5 – Specification of Intra chroma prediction modes and associated names.....	134

	<i>Page</i>
Table 8-6 – Specification of the variable colPic .....	142
Table 8-7 – Specification of PicCodingStruct( X ) .....	142
Table 8-8 – Specification of mbAddrCol, yM, and vertMvScale .....	143
Table 8-9 – Assignment of prediction utilization flags .....	145
Table 8-10 – Derivation of the vertical component of the chroma vector in field coding mode.....	151
Table 8-11 – Differential full-sample luma locations .....	155
Table 8-12 – Assignment of the luma prediction sample predPartLX <sub>L</sub> [ x <sub>L</sub> , y <sub>L</sub> ].....	157
Table 8-13 – Specification of mapping of idx to c <sub>ij</sub> for zig-zag and field scan .....	166
Table 8-14 – Specification of mapping of idx to c <sub>ij</sub> for 8x8 zig-zag and 8x8 field scan .....	168
Table 8-15 – Specification of QP <sub>C</sub> as a function of qp <sub>I</sub> .....	169
Table 8-16 – Derivation of offset dependent threshold variables $\alpha'$ and $\beta'$ from indexA and indexB .....	192
Table 8-17 – Value of variable t' <sub>C0</sub> as a function of indexA and bS.....	194
Table 9-1 – Bit strings with “prefix” and “suffix” bits and assignment to codeNum ranges (informative).....	196
Table 9-2 – Exp-Golomb bit strings and codeNum in explicit form and used as ue(v) (informative) .....	196
Table 9-3 – Assignment of syntax element to codeNum for signed Exp-Golomb coded syntax elements se(v).....	197
Table 9-4 – Assignment of codeNum to values of coded_block_pattern for macroblock prediction modes.....	198
Table 9-5 – coeff_token mapping to TotalCoeff( coeff_token ) and TrailingOnes( coeff_token ) .....	203
Table 9-6 – Codeword table for level_prefix (informative).....	206
Table 9-7 – total_zeros tables for 4x4 blocks with TotalCoeff( coeff_token ) 1 to 7 .....	207
Table 9-8 – total_zeros tables for 4x4 blocks with TotalCoeff( coeff_token ) 8 to 15 .....	208
Table 9-9 – total_zeros tables for chroma DC 2x2 and 2x4 blocks .....	208
Table 9-10 – Tables for run_before.....	209
Table 9-11 – Association of ctxIdx and syntax elements for each slice type in the initialisation process .....	211
Table 9-12 – Values of variables m and n for ctxIdx from 0 to 10 .....	212
Table 9-13 – Values of variables m and n for ctxIdx from 11 to 23 .....	213
Table 9-14 – Values of variables m and n for ctxIdx from 24 to 39 .....	213
Table 9-15 – Values of variables m and n for ctxIdx from 40 to 53 .....	213
Table 9-16 – Values of variables m and n for ctxIdx from 54 to 59, and 399 to 401.....	214
Table 9-17 – Values of variables m and n for ctxIdx from 60 to 69 .....	214
Table 9-18 – Values of variables m and n for ctxIdx from 70 to 104 .....	215
Table 9-19 – Values of variables m and n for ctxIdx from 105 to 165 .....	216
Table 9-20 – Values of variables m and n for ctxIdx from 166 to 226 .....	217
Table 9-21 – Values of variables m and n for ctxIdx from 227 to 275 .....	218
Table 9-22 – Values of variables m and n for ctxIdx from 277 to 337 .....	219
Table 9-23 – Values of variables m and n for ctxIdx from 338 to 398 .....	220
Table 9-24 – Values of variables m and n for ctxIdx from 402 to 459 .....	221
Table 9-25 – Syntax elements and associated types of binarization, maxBinIdxCtx, and ctxIdxOffset.....	223

	<i>Page</i>
Table 9-26 – Bin string of the unary binarization (informative).....	224
Table 9-27 – Binarization for macroblock types in I slices .....	227
Table 9-28 – Binarization for macroblock types in P, SP, and B slices.....	228
Table 9-29 – Binarization for sub-macroblock types in P, SP, and B slices.....	229
Table 9-30 – Assignment of ctxIdxInc to binIdx for all ctxIdxOffset values except those related to the syntax elements coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1.....	231
Table 9-31 – Assignment of ctxIdxBlockCatOffset to ctxBlockCat for syntax elements coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1 .....	232
Table 9-32 – Specification of ctxIdxInc for specific values of ctxIdxOffset and binIdx .....	239
Table 9-33 – Specification of ctxBlockCat for the different blocks .....	240
Table 9-34 – Mapping of scanning position to ctxIdxInc for ctxBlockCat == 5 .....	241
Table 9-35 – Specification of rangeTabLPS depending on pStateIdx and qCodIRangeIdx .....	245
Table 9-36 – State transition table .....	246
Table A-1 – Level limits .....	260
Table A-2 – Specification of cpbBrVclFactor and cpbBrNalFactor .....	262
Table A-3 – Baseline profile level limits .....	263
Table A-4 – Main, High, High 10, High 4:2:2, or High 4:4:4 profile level limits .....	264
Table A-5 – Extended profile level limits.....	265
Table A-6 – Maximum frame rates (frames per second) for some example frame sizes.....	266
Table D-1 – Interpretation of pic_struct .....	295
Table D-2 – Mapping of ct_type to source picture scan .....	296
Table D-3 – Definition of counting_type values.....	296
Table D-4 – scene_transition_type values. ....	303
Table D-5 – model_id values.....	311
Table D-6 – blending_mode_id values .....	312
Table E-1 – Meaning of sample aspect ratio indicator .....	322
Table E-2 – Meaning of video_format.....	323
Table E-3 – Colour primaries .....	324
Table E-4 – Transfer characteristics .....	325
Table E-5 – Matrix coefficients .....	327
Table E-6 – Divisor for computation of $\Delta t_{fi,dpb}(n)$ .....	329
Table F-1 – Organisations providing patent rights licensing notices.....	335

## 0 Introduction

This clause does not form an integral part of this Recommendation | International Standard.

### 0.1 Prologue

This subclause does not form an integral part of this Recommendation | International Standard.

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.

### 0.2 Purpose

This subclause does not form an integral part of this Recommendation | International Standard.

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 subclause does not form an integral part of this Recommendation | International Standard.

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 International Standard

This subclause does not form an integral part of this Recommendation | International Standard.

This International Standard has been jointly developed by ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group. It is published as technically-aligned twin text in both organizations ITU-T and ISO/IEC.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 1 refers to the first (2003) 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 (the current specification) refers to the integrated text containing the first technical corrigendum (2004), the first amendment (the "Fidelity range extensions"), and an additional technical corrigendum (2005). In the ITU-T, the next published version after version 2 was version 4 (due to the completion of the drafting work for version 4 prior to the approval opportunity for a final version 3 text).

## 0.5 Profiles and levels

This subclause does not form an integral part of this Recommendation | International Standard.

This Recommendation | International Standard 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 International Standard, 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 International Standard will facilitate video data interchange among different applications.

Considering the practicality of implementing the full syntax of this International Standard, 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 Recommendation | International Standard. 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 Recommendation | International Standard 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

This subclause does not form an integral part of this Recommendation | International Standard.

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 profile 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 Recommendation | International Standard) 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 quantised, 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 quantised transform coefficient information and encoded using either variable length codes or arithmetic coding.

### 0.6.1 Predictive coding

This subclause does not form an integral part of this Recommendation | International Standard.

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.

### **0.6.2 Coding of progressive and interlaced video**

This subclause does not form an integral part of this Recommendation | International Standard.

This Recommendation | International Standard 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.

### **0.6.3 Picture partitioning into macroblocks and smaller partitions**

This subclause does not form an integral part of this Recommendation | International Standard.

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 Recommendation | International Standard 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 Recommendation | International Standard.

### **0.6.4 Spatial redundancy reduction**

This subclause does not form an integral part of this Recommendation | International Standard.

Both source pictures and prediction residuals have high spatial redundancy. This Recommendation | International Standard 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 quantised. After quantisation 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 quantisation in the encoder are not specified in this Recommendation | International Standard.

## **0.7 How to read this specification**

This subclause does not form an integral part of this Recommendation | International Standard.

It is suggested that the reader start with Clause 1 (Scope) and moves on to clause 3 (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 7.1-7.3 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 specification, the reader should refer to Clauses 2 (Normative references), 4 (Abbreviations), and 5 (Conventions) as needed. Annexes A through E also form an integral part of this Recommendation | International Standard.

Annex A specifies seven profiles (Baseline, Main, Extended, High, High 10, High 4:2:2 and High 4:4:4), 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. Finally, Annex E specifies syntax and semantics of the video usability information parameters of the sequence parameter set.

Throughout this specification, statements appearing with the preamble "NOTE -" are informative and are not an integral part of this Recommendation | International Standard.



# Information technology — Coding of audio-visual objects —

## Part 10: Advanced Video Coding

### 1 Scope

This document specifies ITU-T Recommendation H.264 | ISO/IEC International Standard ISO/IEC 14496-10 video coding.

### 2 Normative references

The following Recommendations and International Standards contain provisions that, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardisation Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

- ITU-T Recommendation T.35 (2000), *Procedure for the allocation of ITU-T defined codes for non-standard facilities*
- ISO/IEC 11578:1996, *Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)*
- ISO/CIE 10527:1991, *CIE standard colorimetric observers*

### 3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

- 3.1 access unit:** A set of *NAL units* always containing exactly one *primary coded picture*. In addition to the *primary coded picture*, an access unit may also contain one or more *redundant coded pictures* 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*.
- 3.2 AC transform coefficient:** Any *transform coefficient* for which the *frequency index* in one or both dimensions is non-zero.
- 3.3 adaptive binary arithmetic decoding process:** An entropy *decoding process* that derives the values of *bins* from a *bitstream* produced by an *adaptive binary arithmetic encoding process*.
- 3.4 adaptive binary arithmetic encoding process:** An entropy *encoding process*, not normatively specified in this Recommendation | International Standard, that codes a sequence of *bins* and produces a *bitstream* that can be decoded using the *adaptive binary arithmetic decoding process*.
- 3.5 alpha blending:** A process not specified by this Recommendation | International Standard, in which an *auxiliary coded picture* is used in combination with a *primary coded picture* and with other data not specified by this Recommendation | International Standard in the *display process*. 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*.