ISO/IEC JTC 1/SC 29

Date: 2008-10-5

ISO/IEC 14496-10:2008/FDAM 1:2008(E)

ISO/IEC JTC 1/SC 29/WG 11

Secretariat:

Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding, AMENDMENT 1: Multiview Video Coding

Document type: International Standard
Document subtype: Amendment
Document stage: (50) Approval
Document language: E

STD Version 2.1c2

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

Amendment 1 to ISO/IEC 14496-10:2008 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding, AMENDMENT 1: Multiview Video Coding

In 0.6, add the following paragraph after the paragraph that starts with "Scalable video coding":

Multiview video coding is specified in Annex H 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 specification. 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.

In 0.7, add the following paragraph after the paragraph that starts with "Annex G specifies":

Annex H specifies multiview video coding (MVC). The reader is referred to Annex H for the entire decoding process for MVC, which is specified there with references being made to clauses 2-9 and Annexes A-E. Subclause H.10 specifies one profile for MVC (Multiview High).

In 7.3.1, NAL unit syntax, replace:

nal_unit_header_svc_extension( ) /* specified in Annex G */

with:

nal_unit_header_extension( ) /* specified in Annexes G and H */

In 7.3.2.1.1, Sequence parameter set data syntax, replace the syntax table with:

seq_parameter_set_data( ) {	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
constraint_set4_flag	0	u(1)
reserved_zero_3bits /* equal to 0 */	0	u(3)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc = = 100 | | profile_idc = = 110 | | profile_idc = = 122 | | profile_idc = = 244 | | profile_idc = = 44 | | profile_idc = = 83 | | profile_idc = = 86 | | profile_idc = = 118) {		
chroma_format_idc	0	ue(v)
if(chroma_format_idc = = 3)		
separate_colour_plane_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v)

bit_depth_chroma_minus8	0	ue(v)
qpprime_y_zero_transform_bypass_flag	0	u(1)
seq_scaling_matrix_present_flag	0	u(1)
if(seq_scaling_matrix_present_flag)		
for(i = 0; i < ((chroma_format_idc != 3) ? 8 : 12); i++) {		
seq_scaling_list_present_flag[i **]**	0	u(1)
if(seq_scaling_list_present_flag[i])		
if(i < 6)		
scaling_list(ScalingList4x4[i], 16, UseDefaultScalingMatrix4x4Flag[i])	0	
else		
scaling_list(ScalingList8x8[i − 6], 64, UseDefaultScalingMatrix8x8Flag[i − 6])	0	
}		
}		
log2_max_frame_num_minus4	0	ue(v)
pic_order_cnt_type	0	ue(v)
if(pic_order_cnt_type = = 0)		
log2_max_pic_order_cnt_lsb_minus4	0	ue(v)
else if(pic_order_cnt_type = = 1) {		
delta_pic_order_always_zero_flag	0	u(1)
offset_for_non_ref_pic	0	se(v)
offset_for_top_to_bottom_field	0	se(v)
num_ref_frames_in_pic_order_cnt_cycle	0	ue(v)
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)		
offset_for_ref_frame[i **]**	0	se(v)
}		
max_num_ref_frames	0	ue(v)
gaps_in_frame_num_value_allowed_flag	0	u(1)
pic_width_in_mbs_minus1	0	ue(v)
pic_height_in_map_units_minus1	0	ue(v)
frame_mbs_only_flag	0	u(1)
if(!frame_mbs_only_flag)		
mb_adaptive_frame_field_flag	0	u(1)
direct_8x8_inference_flag	0	u(1)
frame_cropping_flag	0	u(1)
if(frame_cropping_flag) {		
frame_crop_left_offset	0	ue(v)
frame_crop_right_offset	0	ue(v)
frame_crop_top_offset	0	ue(v)
frame_crop_bottom_offset	0	ue(v)
}		
vui_parameters_present_flag	0	u(1)
if(vui_parameters_present_flag)		
vui_parameters( )	0	
}		

In 7.3.3, Slice header syntax, replace the condition:

```
if( nal_unit_type  = =  5 )
```

with:

```
if( IdrPicFlag )
```

In 7.4.1, NAL unit semantics, make the following changes:

Replace the following paragraph:

For coded video sequences conforming to one or more of the profiles specified in Annex A that are decoded using the decoding process specified in clauses 2-9, VCL and non-VCL NAL units are specified in Table 7-1 in the column labelled "Annex A NAL unit type class". For coded video sequences conforming to one or more of the profiles specified in Annex G that are decoded using the decoding process specified in Annex G, VCL and non-VCL NAL units are specified in Table 7-1 in the column labelled "Annex G NAL unit type class". The entry "suffix dependent" for nal_unit_type equal to 14 is specified as follows.

with:

For coded video sequences conforming to one or more of the profiles specified in Annex A that are decoded using the decoding process specified in clauses 2-9, VCL and non-VCL NAL units are specified in Table 7-1 in the column labelled "Annex A NAL unit type class". For coded video sequences conforming to one or more of the profiles specified in Annex G that are decoded using the decoding process specified in Annex G or specified in Annex H that are decoded using the decoding process specified in Annex H, VCL and non-VCL NAL units are specified in Table 7-1 in the column labelled "Annex G and Annex H NAL unit type class". The entry "suffix dependent" for nal_unit_type equal to 14 is specified as follows.

Replace Table 7-1 with:

Table 7-1 – NAL unit type codes, syntax element categories, and NAL unit type classes

nal_unit_type	Content of NAL unit and RBSP syntax structure	C	Annex A NAL unit type class	Annex G and Annex H NAL unit type class
0	Unspecified		non-VCL	non-VCL
1	Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp( )	2, 3, 4	VCL	VCL
2	Coded slice data partition A slice_data_partition_a_layer_rbsp( )	2	VCL	not applicable
3	Coded slice data partition B slice_data_partition_b_layer_rbsp( )	3	VCL	not applicable
4	Coded slice data partition C slice_data_partition_c_layer_rbsp( )	4	VCL	not applicable
5	Coded slice of an IDR picture slice_layer_without_partitioning_rbsp( )	2, 3	VCL	VCL
6	Supplemental enhancement information (SEI) sei_rbsp( )	5	non-VCL	non-VCL
7	Sequence parameter set seq_parameter_set_rbsp( )	0	non-VCL	non-VCL
8	Picture parameter set pic_parameter_set_rbsp( )	1	non-VCL	non-VCL
9	Access unit delimiter access_unit_delimiter_rbsp( )	6	non-VCL	non-VCL
10	End of sequence end_of_seq_rbsp( )	7	non-VCL	non-VCL
11	End of stream end_of_stream_rbsp( )	8	non-VCL	non-VCL
12	Filler data filler_data_rbsp( )	9	non-VCL	non-VCL
13	Sequence parameter set extension seq_parameter_set_extension_rbsp( )	10	non-VCL	non-VCL
14	Prefix NAL unit prefix_nal_unit_rbsp( ) /* specified in Annex G and Annex H */	2	non-VCL	suffix dependent
15	Subset sequence parameter set subset_seq_parameter_set_rbsp( ) /* specified in Annex G and Annex H */	0	non-VCL	non-VCL
16..18	Reserved		non-VCL	non-VCL
19	Coded slice of an auxiliary coded picture without partitioning slice_layer_without_partitioning_rbsp( )	2, 3, 4	non-VCL	non-VCL
20	Coded slice extension slice_layer extension_rbsp( ) /* specified in Annex G and Annex H */	2, 3, 4	non-VCL	VCL
21..23	Reserved		non-VCL	non-VCL

24..31	Unspecified		non-VCL	non-VCL

In 7.4.2.1.1, Sequence parameter set data semantics, make the following changes:

*Add the following text after the paragraph starting with "**constraint_set3_flag**":*

constraint_set4_flag equal to 1 indicates that the coded video sequence obeys all constraints specified in subclause H.10.1. constraint_set4_flag equal to 0 indicates that the coded video sequence may or may not obey all constraints specified in subclause H.10.1.

Replace:

reserved_zero_4bits shall be equal to 0. Other values of reserved_zero_4bits may be specified in the future by ITU-T | ISO/IEC. Decoders shall ignore the value of reserved_zero_4bits.

with:

reserved_zero_3bits shall be equal to 0. Other values of reserved_zero_3bits may be specified in the future by ITU-T | ISO/IEC. Decoders shall ignore the value of reserved_zero_3bits.

In C, Hypothetical reference decoder, replace:

For each picture in the bitstream, the variable OutputFlag for the decoded picture and, when applicable, the reference base picture is set as follows.

– If the coded video sequence containing the picture conforms to one or more of the profiles specified in Annex A and the decoding process specified in clauses 2-9 is used), OutputFlag is set equal to 1.

– Otherwise (the coded video sequence containing the picture conforms to a profile specified in Annex G and the decoding process specified in Annex G is used), the following applies:

  – For a reference base picture, OutputFlag is set equal to 0.

  – For a decoded picture, OutputFlag is set equal to the value of the output_flag syntax element of the target layer representation.

The operation of the CPB is specified in subclause C.1. The instantaneous decoder operation is specified in clauses 2-9 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A, or specified in Annex G when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G. The operation of the DPB is specified in subclause C.2. The output cropping is specified in subclause C.2.2.

HSS and HRD information concerning the number of enumerated delivery schedules and their associated bit rates and buffer sizes is specified in subclauses E.1.1, E.1.2, E.2.1 and E.2.2. The HRD is initialised as specified by the buffering period SEI message as specified in subclauses D.1.1 and D.2.1. The removal timing of access units from the CPB and output timing from the DPB are specified in the picture timing SEI message as specified in subclauses D.1.2 and D.2.2. All timing information relating to a specific access unit shall arrive prior to the CPB removal time of the access unit.

with:

For each picture in the bitstream, the variable OutputFlag for the decoded picture and, when applicable, the reference base picture, is set as follows.

– If the coded video sequence containing the picture conforms to one or more of the profiles specified in Annex A and the decoding process specified in clauses 2-9 is used), OutputFlag is set equal to 1.

– Otherwise if the coded video sequence containing the picture conforms to a profile specified in Annex G and the decoding process specified in Annex G is used, the following applies:

  – For a reference base picture, OutputFlag is set equal to 0.

  – For a decoded picture, OutputFlag is set equal to the value of the output_flag syntax element of the target layer representation.

– Otherwise (the coded video sequence containing the picture conforms to a profile specified in Annex H and the decoding process specified in Annex H is used), the following applies:

  – For the decoded view components of the target output views, OutputFlag is set equal to 1.

– For the decoded view components of other views, OutputFlag is set to 0.

The operation of the CPB is specified in subclause C.1. The instantaneous decoder operation is specified in clauses 2-9 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A, or specified in Annex G when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G, or specified in Annex H when decoding a coded video sequence conforming to one or more of the profiles specified in Annex H. The operation of the DPB is specified in subclause C.2. The output cropping is specified in subclause C.2.2.

HSS and HRD information concerning the number of enumerated delivery schedules and their associated bit rates and buffer sizes is specified in subclauses E.1.1, E.1.2, E.2.1 and E.2.2 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or specified in subclauses G.14.1 and G.14.2 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G, or specified in subclauses H.14.1 and H.14.2 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H. The HRD is initialised as specified by the buffering period SEI message as specified in subclauses D.1.1 and D.2.1. The removal timing of access units from the CPB and output timing from the DPB are specified in the picture timing SEI message as specified in subclauses D.1.2 and D.2.2. All timing information relating to a specific access unit shall arrive prior to the CPB removal time of the access unit.

In C.2, Operation of the decoded picture buffer (DPB), add the following text after the sentence, "When decoding a coded video sequence conforming to one or more of the profiles specified in Annex G":

When decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H, each of the frame buffers may contain a decoded frame view component, a decoded complementary field view component pair, or a single (non-paired) decoded field view component that is marked as "used for reference" (reference pictures) or is held for future output (reordered or delayed pictures).

In C.2.1, Decoding of gaps in frame_num and storage of "non-existing" frames, add the following paragraph at the start of the subclause:

When decoding a coded video sequence conforming to a profile specified in Annex H using the decoding process specified in Annex H, the following process in this subclause is repeatedly invoked for each view in increasing order of view order index, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component". During the invocation of the process for a particular view, only view components of the particular view may be removed from the DPB.

In C.2.2, Picture decoding and output, make the following changes:

Add the following text at the start of the subclause:

When the coded video sequence containing the picture conforms to a profile specified in Annex H and the decoding process specified in Annex H is used, the following applies:

– When at least one view component of picture n has OutputFlag equal to 1, it is considered that picture n has OutputFlag equal to 1.

– When at least one view component of picture n is output, it is considered that picture n is output.

– When at least one view component of picture n is stored in DPB, it is considered that picture n is stored in the DPB.

Replace:

NOTE – When the current picture is a reference picture it will be stored in the DPB.

with:

NOTE – When the current picture or a view component of the current picture has nal_ref_idc greater than 0, it will be stored in the DPB.

Replace:

When output, the picture shall be cropped, using the cropping rectangle specified in the active sequence parameter set for the picture.

with:

When the coded video sequence containing the picture conforms to a profile specified in Annex H and the decoding process specified in Annex H is used, and picture n is output, the view components of all the target output views in picture n are output at the same time instant and in increasing order of VOIdx.

When output, the picture or a view component of the picture shall be cropped, using the cropping rectangle specified in the active sequence parameter set for the picture or the view component.

In C.2.3 Removal of pictures from the DPB before possible insertion of the current picture, make the following changes:

Add the following at the start of the subclause:

When decoding a coded video sequence conforming to a profile specified in Annex H using the decoding process specified in Annex H, the following process in this subclause is repeatedly invoked for each view in view decoding order, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component". During the invocation of the process for a particular view, only view components of the particular view may be removed from the DPB.

Replace:

All reference pictures in the DPB are marked as "unused for reference" as specified in subclause 8.2.5.1 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or as specified in subclause G.8.2.4 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G.

with:

All reference pictures in the DPB are marked as "unused for reference" as specified in subclause 8.2.5.1 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or as specified in subclause G.8.2.4 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G, or as specified in subclause H.8.3 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H.

Replace:

Otherwise (the slice header of the current picture does not include memory_management_control_operation equal to 5), the decoded reference picture marking process specified in subclause 8.2.5 is invoked when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or the decoded reference picture marking process specified in subclause G.8.2.4 is invoked when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G.

with:

Otherwise (the slice header of the current picture does not include memory_management_control_operation equal to 5), the decoded reference picture marking process specified in subclause 8.2.5 is invoked when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or the decoded reference picture marking process specified in subclause G.8.2.4 is invoked when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G, or the decoded reference picture marking process specified in subclause H.8.3 is involved when decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H.

In C.2.4 Current decoded picture marking and storage, add the following at the start of the subclause:

When decoding a coded video sequence conforming to a profile specified in Annex H using the decoding process specified in Annex H, the following process in this subclause is repeatedly invoked for each view in increasing order of view order index, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component". During the invocation of the process for a particular view, only view components of the particular view may be removed from the DPB.

In C.3, Bitstream Conformance, make the following changes:

Replace:

The nominal removal times of pictures from the CPB (starting from the second picture in decoding order), shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclauses A.3.1 through A.3.3 for the profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, and they shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclauses G.10.2.1 and G.10.2.2 for profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G.

with:

The nominal removal times of pictures from the CPB (starting from the second picture in decoding order), shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclauses A.3.1 through A.3.3 for the profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, and they shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclauses G.10.2.1 and G.10.2.2 for profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G, and they shall satisfy the constraints on $t_{r,n}(n)$ and $t_r(n)$ expressed in subclause H.10.2 for the profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H.

Replace:

The value of $\Delta_{to,dpb}(n)$ as given by Equation C-13, which is the difference between the output time of a picture and that of the first picture following it in output order and having OutputFlag equal to 1, shall satisfy the constraint expressed in subclause A.3.1 for the profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, and it shall satisfy the constraint expressed in subclause G.10.2.1 for profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G.

with:

The value of $\Delta_{to,dpb}(n)$ as given by Equation C-13, which is the difference between the output time of a picture and that of the first picture following it in output order and having OutputFlag equal to 1, shall satisfy the constraint expressed in subclause A.3.1 for the profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, and it shall satisfy the constraint expressed in subclause G.10.2.1 for profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G, and it shall satisfy the constraints expressed in subclauses H.10.2 for the profile and level specified in the bitstream when decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H.

In C.4.1, Operation of the output order DPB, add the following text after the sentence starting with, "When decoding a coded video sequence conforming to one or more of the profiles specified in Annex G":

When decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H, each of the frame buffers may contain a decoded frame view component, a decoded complementary field view component pair, or a single (non-paired) decoded field view component that is marked as "used for reference" (reference pictures) or is held for future output (reordered or delayed pictures).

In C.4.2, Decoding of gaps in frame_num and storage of "non-existing" pictures, add the following at the start of the subclause:

When decoding a coded video sequence conforming to a profile specified in Annex H using the decoding process specified in Annex H, the following process in this subclause is repeatedly invoked for each view in increasing order of view order index, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component". During the invocation of the process for a particular view, only view components of the particular view may be removed from the DPB.

In C.4.4, Removal of pictures from the DPB before possible insertion of the current picture, make the following changes:

Add the following at the start of the subclause:

When decoding a coded video sequence conforming to a profile specified in Annex H using the decoding process specified in Annex H, the following process in this subclause is repeatedly invoked for each view in inreasing order of view order index, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component". During the invocation of the process for a particular view, only view components of the particular view may be removed from the DPB.

Replace:

All reference pictures in the DPB are marked as "unused for reference" as specified in subclause 8.2.5 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or as specified in subclause G.8.2.4 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G.

with:

All reference pictures in the DPB are marked as "unused for reference" as specified in subclause 8.2.5 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or as specified in subclause G.8.2.4 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G, or as specified in subclause H.8.3 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H.

Replace:

Otherwise (the decoded picture is not an IDR picture), the decoded reference picture marking process is invoked as specified in subclause 8.2.5 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or as specified in subclause G.8.2.4 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G.

with:

Otherwise (the decoded picture is not an IDR picture), the decoded reference picture marking process is invoked as specified in subclause 8.2.5 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex A using the decoding process specified in clauses 2-9, or as specified in subclause G.8.2.4 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex G using the decoding process specified in Annex G, or as specified in subclause H.8.3 when decoding a coded video sequence conforming to one or more of the profiles specified in Annex H using the decoding process specified in Annex H.

In C.4.5, Current decoded picture marking and storage, add the following at the start of the subclause:

When decoding a coded video sequence conforming to a profile specified in Annex H using the decoding process specified in Annex H, the following process in this subclause is repeatedly invoked for each view in increasing order of view order index, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component". During the invocation of the process for a particular view, only view components of the particular view may be removed from the DPB.

In D.1, SEI payload syntax, replace the syntax table with:

sei_payload(payloadType, payloadSize) {	C	Descriptor
if(payloadType = = 0)		
buffering_period(payloadSize)	5	
else if(payloadType = = 1)		
pic_timing(payloadSize)	5	
else if(payloadType = = 2)		
pan_scan_rect(payloadSize)	5	
else if(payloadType = = 3)		
filler_payload(payloadSize)	5	
else if(payloadType = = 4)		
user_data_registered_itu_t_t35(payloadSize)	5	
else if(payloadType = = 5)		
user_data_unregistered(payloadSize)	5	
else if(payloadType = = 6)		
recovery_point(payloadSize)	5	
else if(payloadType = = 7)		
dec_ref_pic_marking_repetition(payloadSize)	5	
else if(payloadType = = 8)		
spare_pic(payloadSize)	5	
else if(payloadType = = 9)		
scene_info(payloadSize)	5	
else if(payloadType = = 10)		
sub_seq_info(payloadSize)	5	
else if(payloadType = = 11)		
sub_seq_layer_characteristics(payloadSize)	5	
else if(payloadType = = 12)		
sub_seq_characteristics(payloadSize)	5	
else if(payloadType = = 13)		
full_frame_freeze(payloadSize)	5	
else if(payloadType = = 14)		
full_frame_freeze_release(payloadSize)	5	
else if(payloadType = = 15)		
full_frame_snapshot(payloadSize)	5	
else if(payloadType = = 16)		
progressive_refinement_segment_start(payloadSize)	5	
else if(payloadType = = 17)		
progressive_refinement_segment_end(payloadSize)	5	
else if(payloadType = = 18)		
motion_constrained_slice_group_set(payloadSize)	5	
else if(payloadType = = 19)		
film_grain_characteristics(payloadSize)	5	
else if(payloadType = = 20)		
deblocking_filter_display_preference(payloadSize)	5	
else if(payloadType = = 21)		
stereo_video_info(payloadSize)	5	

else if(payloadType = = 22)		
post_filter_hint(payloadSize)	5	
else if(payloadType = = 23)		
tone_mapping_info(payloadSize)	5	
else if(payloadType = = 24)		
scalability_info(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 25)		
sub_pic_scalable_layer(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 26)		
non_required_layer_rep(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 27)		
priority_layer_info(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 28)		
layers_not_present(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 29)		
layer_dependency_change(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 30)		
scalable_nesting(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 31)		
base_layer_temporal_hrd(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 32)		
quality_layer_integrity_check(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 33)		
redundant_pic_property(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 34)		
tl0_picture_index(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 35)		
tl_switching_point(payloadSize) /* specified in Annex G */	5	
else if(payloadType = = 36)		
parallel_decoding_info(payloadSize) /* specified in Annex H */	5	
else if(payloadType = = 37)		
mvc_scalable_nesting(payloadSize) /* specified in Annex H */	5	
else if(payloadType = = 38)		
view_scalability_info(payloadSize) /* specified in Annex H */	5	
else if(payloadType = = 39)		
multiview_scene_info(payloadSize) /* specified in Annex H */	5	
else if(payloadType = = 40)		
multiview_acquisition_info(payloadSize) /* specified in Annex H */	5	
else if(payloadType = = 41)		
non_required_view_component(payloadSize) /* specified in Annex H */	5	
else if(payloadType = = 42)		
view_dependency_change(payloadSize) /* specified in Annex H */	5	
else if(payloadType = = 43)		
operation_points_not_present(payloadSize) /* specified in Annex H */	5	
else if(payloadType = = 44)		
base_view_temporal_hrd(payloadSize) /* specified in Annex H */	5	
else		

reserved_sei_message(payloadSize)	5	
if(!byte_aligned( )) {		
bit_equal_to_one /* equal to 1 */	5	f(1)
while(!byte_aligned( ))		
bit_equal_to_zero /* equal to 0 */	5	f(1)
}		
}		

In G.3.8, replace "coded slice in scalable extension" with "coded slice extension".

In G.7.3.1.1, replace the heading "NAL unit header SVC extension syntax" with "NAL unit header extension syntax".

In G.7.3.2.13, replace the heading "Slice layer in scalable extension RBSP syntax" with "Slice layer extension RBSP syntax".

In G.7.4.1.1, replace the heading "NAL unit header SVC extension semantics"with "NAL unit header extension semantics".

In G.7.4.1.1, replace "coded slice in scalable extension" with "coded slice extension".

In G.7.4.1.2.1, replace each of the two instances of "coded slice in scalable extension" with "coded slice extension".

Add a new Annex H, Multiview video coding, with the following text:

Annex H
Multiview video coding

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies multiview video coding, referred to as MVC.

H.1 Scope

Bitstreams and decoders conforming to the profile specified in this annex are completely specified in this annex with reference made to clauses 2-9 and Annexes A-E.

H.2 Normative References

The specifications in clause 2 apply with the following additions:

– IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems*.

H.3 Definitions

For the purpose of this annex, the following definitions apply in addition to the definitions in clause 3. These definitions are either not present in clause 3 or replace definitions in clause 3.

H.3.1 **access unit**: A set of *NAL units* that are consecutive in *decoding order* and contain exactly one *primary coded picture* consisting of one or more *view components*. 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*. All *slices* or *slice data partitions* in an access unit have the same value of *picture order count*.

H.3.2 anchor access unit: An *access unit* in which the *primary coded picture* is an *anchor picture*.

H.3.3 anchor picture: A *coded picture* in which all *slices* reference only *slices* within the same *access unit*, i.e., 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. An anchor picture has anchor_pic_flag to 1 for all the *prefix NAL units* and all the slice extension NAL units.

H.3.4 anchor view component: A *view component* in an *anchor picture*. All *view components* in an *anchor picture* are anchor view components.

H.3.5 associated NAL unit: A *NAL unit* that directly succeeds a *prefix NAL unit* in *decoding order*.

H.3.6 base view: A *view* that has the minimum value of *view order index* in a coded video sequence. 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 or 5. The base view bitstream conforms to one or more profiles specified in Annex A. There is only one base view in a *coded video sequence*.

H.3.7 bitstream subset: A *bitstream* that is derived as a *subset* from a *bitstream* by discarding zero or more *NAL units*. A *bitstream subset* is also referred to as a *sub-bitstream*.

H.3.8 decoded view component: A *decoded view component* is derived by decoding a *view component*. A *decoded view component* is either a decoded *frame view component*, or a decoded *field view component*.

H.3.9 direct prediction: An *inter prediction* or *inter-view prediction* for a block for which no *motion vector* is decoded. Two direct prediction modes are specified that are referred to as spatial direct prediction mode and temporal direct prediction mode.

H.3.10 field view component: A *view component* of a *field*.

H.3.11 frame view component: A *view component* of a *frame*.

H.3.12 instantaneous decoding refresh (IDR) picture: A *coded picture* in which all *slices* have IdrPicFlag equal to 1 that causes the *decoding process* to mark all *reference pictures* as "unused for reference" immediately after decoding the IDR picture. After the decoding of an IDR picture all following *coded pictures* in *decoding order* can be decoded without *inter prediction* from any *picture* decoded prior to the IDR picture. The first *picture* of each *coded video sequence* is an IDR picture.

H.3.13 instantaneous decoding refresh (IDR) view component: A *view component* in an *IDR picture*. All *view components* in an *IDR picture* are IDR view components.

H.3.14 inter-view coding: Coding of a block, macroblock, slice, or picture that uses inter-view prediction.

H.3.15 inter-view only reference component: A *decoded view component* with nal_ref_idc equal to 0 and inter_view_flag equal to 1. An inter-view only reference component is used for *inter-view prediction* in the *decoding process* of subsequent *view components* in *decoding order*, but is not used for *inter prediction* by any *view components*. Inter-view only reference components are *non-reference pictures*.

H.3.16 inter-view prediction: A *prediction* derived from decoded samples of *inter-view reference pictures* or *inter-view only reference components* for decoding another *view component* in the same access unit.

H.3.17 inter-view prediction reference: A collective term for *inter-view reference picture* or *inter-view only reference components*.

H.3.18 inter-view reference index: An index into a list of *view components* for *inter-view prediction* in an initialised *reference picture list* in decoding all the *anchor view components* or all the *non-anchor view components* of any particular view as specified in the *sequence parameter set* MVC extension.

H.3.19 inter-view reference picture: A *decoded view component* with nal_ref_idc greater than 0 and inter_view_flag equal to 1. An *inter-view reference picture* contains samples that may be used for *inter prediction* and *inter-view prediction* in the *decoding process* of subsequent *pictures* in *decoding order*. Inter-view reference pictures are *reference pictures*.

H.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)*.

H.3.21 macroblock partition: A *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*.

H.3.22 motion vector: A 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*.

H.3.23 MVC sequence parameter set: A collective term for *sequence parameter set* or *subset sequence parameter set*.

H.3.24 MVC sequence parameter set RBSP: A collective term for sequence parameter set RBSP or subset sequence parameter set RBSP.

H.3.25 non-anchor access unit: An *access unit* that is not an *anchor access unit*.

H.3.26 non-anchor picture: A *coded picture* that is not an *anchor picture*.

H.3.27 non-anchor view component: A *view component* that is not an *anchor view component*.

H.3.28 non-base view: A *view* that is not the *base view*. *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*.

H.3.29 non-reference picture: A *view component* coded with nal_ref_idc equal to 0. A non-reference picture is not used for *inter prediction* in decoding any other view components.

H.3.30 operation point: An 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*. 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 H.8.5.3 with tIdTarget 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*.

H.3.31 picture order count: A variable that applies to each view independently having a value that is non-decreasing (except for wrap-around) with increasing view component position in output order relative to the previous *IDR view component* in *decoding order* or relative to the previous *view component* containing the *memory management control operation* that marks all *reference pictures* in the same view as "unused for reference".

H.3.32 prefix NAL unit: A *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. The *NAL unit* that immediately succeeds 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*.

H.3.33 reference picture: A *view component* coded with nal_ref_idc greater than 0. A reference picture may be used for *inter prediction* or *inter-view prediction* in decoding the following *view components* in *decoding order*.

H.3.34 reference picture list: A list of *reference pictures* and *inter-view only reference components* that are used for *inter prediction* or *inter-view prediction* of a *P*, *B*, or *SP slice*. 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.

H.3.35 reference picture list 0: A *reference picture list* used for *inter prediction* or *inter-view prediction* of a *P*, *B*, or *SP slice*. All inter prediction or inter-view 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*.

H.3.36 reference picture list 1: A *reference picture list* used for *inter prediction* or *inter-view prediction* of a *B slice*. 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*.

H.3.37 reference picture marking: Specifies the means by which *decoded view components* are marked for *inter prediction* or *inter-view prediction*.

H.3.38 sub-bitstream: A *subset* of a *bitstream*. A *sub-bitstream* is also referred to as a *bitstream subset*.

H.3.39 subset: A subset contains only elements that are also contained in the set from which the subset is derived. The subset may be identical to the set from which it is derived.

H.3.40 target output view: A view that is to be output. 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 – The output views may be requested by a receiver and may be negotiated between the receiver and the sender.

H.3.41 target temporal level: The *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*.

H.3.42 view: A sequence of *view components* associated with an identical value of view_id.

H.3.43 view component: A *coded representation* of a view in a single *access unit*.

H.3.44 view order index: An index that indicates the decoding order of *view components* in an *access unit*.

H.4 Abbreviations

The specifications in clause 4 apply in addition to the following:

MVC Multiview Video Coding

H.5 Conventions

The specifications in clause 5 apply.

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

The specifications in clause 6 apply with substitution of MVC sequence parameter set for sequence parameter set.

H.7 Syntax and semantics

This clause specifies syntax and semantics for coded video sequences that conform to one or more of the profiles specified in this annex.

H.7.1 Method of specifying syntax in tabular form

The specifications in subclause 7.1 apply.

H.7.2 Specification of syntax functions, categories, and descriptors

The specifications in subclauses 7.2 apply.

H.7.3 Syntax in tabular form

H.7.3.1 NAL unit syntax

The syntax table is specified in subclause 7.3.1.

H.7.3.1.1 NAL unit header extension syntax

nal_unit_header_extension( ) {	C	Descriptor
reserved_zero_bit	All	u(1)
idr_flag	All	u(1)
priority_id	All	u(6)
view_id	All	u(10)
temporal_id	All	u(3)
anchor_pic_flag	All	u(1)
inter_view_flag	All	u(1)
reserved_one_bit	All	u(1)
}		

H.7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax

H.7.3.2.1 Sequence parameter set RBSP syntax

The syntax table is specified in subclause 7.3.2.1.

H.7.3.2.1.1 Sequence parameter set data syntax

The syntax table is specified in subclause 7.3.2.1.1.

H.7.3.2.1.1.1 Scaling list syntax

The syntax table is specified in subclause 7.3.2.1.1.1.

H.7.3.2.1.2 Sequence parameter set extension RBSP syntax

The syntax table is specified in subclause 7.3.2.1.2.

H.7.3.2.1.3 Subset sequence parameter set RBSP syntax

subset_seq_parameter_set_rbsp( ) {	C	Descriptor
seq_parameter_set_data( )	0	
if(profile_idc = = 118) {		
bit_equal_to_one /* equal to 1 */	0	f(1)
seq_parameter_set_mvc_extension( )	0	
mvc_vui_parameters_present_flag	0	u(1)
if(mvc_vui_parameters_present_flag = = 1)		
mvc_vui_parameters_extension( )	0	
}		
additional_extension2_flag	0	u(1)
if(additional_extension2_flag = = 1)		
while(more_rbsp_data( ))		
additional_extension2_data_flag	0	u(1)
rbsp_trailing_bits( )		
}		

H.7.3.2.1.4 Sequence parameter set MVC extension syntax

seq_parameter_set_mvc_extension( ) {	C	Descriptor
num_views_minus1	0	ue(v)
for(i = 0; i <= num_views_minus1; i++)		
view_id[i **]**	0	ue(v)
for(i = 1; i <= num_views_minus1; i++) {		
num_anchor_refs_l0[i **]**	0	ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i **][** j **]**	0	ue(v)
num_anchor_refs_l1[i **]**	0	ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i **][** j **]**	0	ue(v)
}		
for(i = 1; i <= num_views_minus1; i++) {		
num_non_anchor_refs_l0[i **]**	0	ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		

non_anchor_ref_l0[i **][** j **]**	0	ue(v)
num_non_anchor_refs_l1[i **]**	0	ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i **][** j **]**	0	ue(v)
}		
num_level_values_signalled	0	ue(v)
for(i = 0; i < num_level_values_signalled; i++) {		
level_idc[i **]**	0	u(8)
num_applicable_ops_minus1[i **]**	0	ue(v)
for(j = 0; j <= num_applicable_ops_minus1[i]; j++) {		
applicable_op_temporal_id[i **][** j **]**	0	u(3)
applicable_op_num_target_views_minus1[i **][** j **]**	0	ue(v)
for(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++)		
applicable_op_target_view_id[i **][** j **][** k **]**	0	ue(v)
applicable_op_num_views_minus1[i **][** j **]**	0	ue(v)
}		
}		
}		

H.7.3.2.2 Picture parameter set RBSP syntax

The syntax table is specified in subclause 7.3.2.2.

H.7.3.2.3 Supplemental enhancement information RBSP syntax

The syntax table is specified in subclause 7.3.2.3.

H.7.3.2.3.1 Supplemental enhancement information message syntax

The syntax table is specified in subclause 7.3.2.3.1.

H.7.3.2.4 Access unit delimiter RBSP syntax

The syntax table is specified in subclause 7.3.2.4.

H.7.3.2.5 End of sequence RBSP syntax

The syntax table is specified in subclause 7.3.2.5.

H.7.3.2.6 End of stream RBSP syntax

The syntax table is specified in subclause 7.3.2.6.

H.7.3.2.7 Filler data RBSP syntax

The syntax table is specified in subclause 7.3.2.7.

H.7.3.2.8 Slice layer without partitioning RBSP syntax

The syntax table is specified in subclause 7.3.2.8.

H.7.3.2.9 Slice data partition RBSP syntax

Slice data partition syntax is not present in coded video sequences conforming to one or more of the profiles specified in this Annex.

H.7.3.2.10 RBSP slice trailing bits syntax

The syntax table is specified in subclause 7.3.2.10.

H.7.3.2.11 RBSP trailing bits syntax

The syntax table is specified in subclause 7.3.2.11.

H.7.3.2.12 Prefix NAL unit RBSP syntax

prefix_nal_unit_rbsp( ) {	C	Descriptor
}		

H.7.3.2.13 Slice layer extension RBSP syntax

slice_layer_extension_rbsp( ) {	C	Descriptor
slice_header( )	2	
slice_data( )	2 | 3 | 4	
rbsp_slice_trailing_bits( )	2	
}		

H.7.3.3 Slice header syntax

The syntax table is specified in subclause 7.3.3.

H.7.3.3.1 Reference picture list modification syntax

ref_pic_list_modification( ) {	C	Descriptor
if(slice_type % 5 != 2 && slice_type != 4) {		
ref_pic_list_modification_flag_l0	2	u(1)
if(ref_pic_list_modification_flag_l0)		
do {		
modification_of_pic_nums_idc	2	ue(v)
if(modification_of_pic_nums_idc = = 0 || modification_of_pic_nums_idc = = 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(modification_of_pic_nums_idc = = 2)		
long_term_pic_num	2	ue(v)
else if (modification_of_pic_nums_idc = = 4 || modification_of_pic_nums_idc = = 5)		
abs_diff_view_idx_minus1	2	ue(v)
} while(modification_of_pic_nums_idc != 3)		
}		
if(slice_type % 5 = = 1) {		
ref_pic_list_modification_flag_l1	2	u(1)
if(ref_pic_list_modification_flag_l1)		
do {		
modification_of_pic_nums_idc	2	ue(v)
if(modification_of_pic_nums_idc = = 0 || modification_of_pic_nums_idc = = 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(modification_of_pic_nums_idc = = 2)		
long_term_pic_num	2	ue(v)
else if (modification_of_pic_nums_idc = = 4 || modification_of_pic_nums_idc = = 5)		

abs_diff_view_idx_minus1	2	ue(v)
} while(modification_of_pic_nums_idc != 3)		
}		
}		

H.7.3.3.2 Prediction weight table syntax

The syntax table is specified in subclause 7.3.3.2.

H.7.3.3.3 Decoded reference picture marking syntax

The syntax table is specified in subclause 7.3.3.3.

H.7.3.4 Slice data syntax

The syntax table is specified in subclause 7.3.4.

H.7.3.5 Macroblock layer syntax

The syntax table is specified in subclause 7.3.5.

H.7.3.5.1 Macroblock prediction syntax

The syntax table is specified in subclause 7.3.5.1.

H.7.3.5.2 Sub-macroblock prediction syntax

The syntax table is specified in subclause 7.3.5.2.

H.7.3.5.3 Residual data syntax

The syntax table is specified in subclause 7.3.5.3.

H.7.3.5.3.1 Residual luma syntax

The syntax table is specified in subclause 7.3.5.3.1.

H.7.3.5.3.2 Residual block CAVLC syntax

The syntax table is specified in subclause 7.3.5.3.2.

H.7.3.5.3.3 Residual block CABAC syntax

The syntax table is specified in subclause 7.3.5.3.3.

H.7.4 Semantics

Semantics associated with the syntax structures and syntax elements within these structures (in subclause H.7.3 and in subclause 7.3 by reference in subclause H.7.3) are specified in this subclause and by reference to subclause 7.4. When the semantics of a syntax element are specified using a table or a set of tables, any values that are not specified in the table(s) shall not be present in the bitstream unless otherwise specified in this Recommendation | International Standard.

H.7.4.1 NAL unit semantics

The semantics for the syntax elements in subclause H.7.3.1 are specified in subclause 7.4.1. The following specifications additionally apply.

For NAL units with nal_unit_type equal to 14, nal_ref_idc shall be identical to nal_ref_idc of the associated NAL unit, which succeeds the NAL unit with nal_unit_type equal to 14 in decoding order.

The value of nal_ref_idc shall be identical for all VCL NAL units of a view component.

H.7.4.1.1 NAL unit header extension semantics

The syntax elements idr_flag, priority_id, view_id, temporal_id, anchor_pic_flag, and inter_view_flag, when present in a prefix NAL unit, are considered to apply to the associated NAL unit.

reserved_zero_bit shall be equal to 0 for coded video sequences with profile_idc equal to 118 in bitstreams conforming to this Recommendation | International Standard. Decoders conforming to this Recommendation | International Standard shall ignore the value of reserved_zero_bit when profile_idc is equal to 118.

idr_flag equal to 1 specifies that the current access unit is an IDR access unit.

The value of idr_flag shall be the same for all VCL NAL units of an access unit. When idr_flag is equal to 1 for a prefix NAL unit, the associated NAL unit shall have nal_unit_type equal to 5. When idr_flag is equal to 0 for a prefix NAL unit, the associated NAL unit shall have nal_unit_type equal to 1.

When nal_ref_idc is equal to 0, the value of idr_flag shall be equal to 0.

For NAL units in which idr_flag is present, the variable IdrPicFlag derived in subclause 7.4.1 is modified by setting it equal to idr_flag.

priority_id specifies a priority identifier for the NAL unit. A lower value of priority_id specifies a higher priority. The assignment of values to priority_id is constrained by the sub-bitstream extraction process as specified in subclause H.8.5.3.

NOTE 1 – The syntax element priority_id is not used by the decoding process specified in this Recommendation | International Standard. The syntax element priority_id may be used as determined by the application within the specified constraints.

view_id specifies a view identifier for the NAL unit. NAL units with the same value of view_id belong to the same view. The assignment of values to view_id is constrained by the sub-bitstream extraction process as specified in subclause H.8.5.3.

The variable VOIdx, representing the view order index of the view identified by view_id, is derived as follows:

– VOIdx is set equal to i for which the syntax element view_id[i] included in the referred subset sequence parameter set is equal to view_id.

temporal_id specifies a temporal identifier for the NAL unit.

The value of temporal_id shall be the same for all prefix and coded slice extension NAL units of an access unit. When an access unit contains any NAL unit with nal_unit_type equal to 5 or idr_flag equal to 1, temporal_id shall be equal to 0.

The assignment of values to temporal_id is further constrained by the sub-bitstream extraction process as specified in subclause H.8.5.3.

anchor_pic_flag equal to 1 specifies that the current access unit is an anchor access unit.

When idr_flag is equal to 1, anchor_pic_flag shall be equal to 1.

When nal_ref_idc is equal to 0, the value of anchor_pic_flag shall be equal to 0.

The value of anchor_pic_flag shall be the same for all VCL NAL units of an access unit.

inter_view_flag equal to 0 specifies that the current view component is not used for inter-view prediction by any other view component in the current access unit. inter_view_flag equal to 1 specifies that the current view component may be used for inter-view prediction by other view components in the current access unit.

The value of inter_view_flag shall be the same for all VCL NAL units of a view component.

reserved_one_bit shall be equal to 1. The value of reserved_one_bit may be specified by future extension of this Recommendation | International Standard. Decoders shall ignore the value of reserved_one_bit.

H.7.4.1.2 Order of NAL units and association to coded pictures, access units, and video sequences

This subclause specifies constraints on the order of NAL units in the bitstream. Any order of NAL units in the bitstream obeying these constraints is referred to in the text as the decoding order of NAL units. Within a NAL unit, the syntax in subclauses 7.3, D.1, E.1, H.7.3, H.13.1 and H.14.1 specifies the decoding order of syntax elements. Decoders conforming to this Recommendation | International Standard shall be capable of receiving NAL units and their syntax elements in decoding order.

H.7.4.1.2.1 Order of MVC sequence parameter set RBSPs and picture parameter set RBSPs and their activation

NOTE 1 – The sequence and picture parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded macroblock data. Sequence and picture parameter sets may, in some applications, be conveyed "out-of-band" using a reliable transport mechanism.

A picture parameter set RBSP includes parameters that can be referred to by the coded slice NAL units of one or more view components of one or more coded pictures.

Each picture parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one picture parameter set RBSP is considered as the active picture parameter set RBSP at any given moment during the operation of the decoding process, and when any particular picture parameter set RBSP becomes the active picture parameter set RBSP, the previously-active picture parameter set RBSP (if any) is deactivated.

In addition to the active picture parameter set RBSP, zero or more picture parameter set RBSPs may be specifically active for view components (with a particular value of VOIdx less than VOIdxMax) that may be referred to through inter-view prediction in decoding the view component with VOIdx equal to VOIdxMax. Such a picture parameter set RBSP is referred to as active view picture parameter set RBSP for the particular value of VOIdx. The restrictions on active picture parameter set RBSPs also apply to active view picture parameter set RBSPs for a particular value of VOIdx less than VOIdxMax.

When a picture parameter set RBSP (with a particular value of pic_parameter_set_id) is not the active picture parameter set RBSP and it is referred to by a coded slice NAL unit with VOIdx equal to VOIdxMax (using that value of pic_parameter_set_id), it is activated. This picture parameter set RBSP is called the active picture parameter set RBSP until it is deactivated when another picture parameter set RBSP becomes the active picture parameter set RBSP. A picture parameter set RBSP, with that particular value of pic_parameter_set_id, shall be available to the decoding process prior to its activation.

When a picture parameter set RBSP (with a particular value of pic_parameter_set_id) is not the active view picture parameter set for a particular value of VOIdx less than VOIdxMax and it is referred to by a coded slice NAL unit with the particular value of VOIdx (using that value of pic_parameter_set_id), it is activated for view components with the particular value of VOIdx. This picture parameter set RBSP is called the active view picture parameter set RBSP for the particular value of VOIdx until it is deactivated when another picture parameter set RBSP becomes the active view picture parameter set RBSP for the particular value of VOIdx. A picture parameter set RBSP, with that particular value of pic_parameter_set_id, shall be available to the decoding process prior to its activation.

Any picture parameter set NAL unit containing the value of pic_parameter_set_id for the active picture parameter set RBSP for a coded picture shall have the same content as that of the active picture parameter set RBSP for this coded picture unless it follows the last VCL NAL unit of this coded picture and precedes the first VCL NAL unit of another coded picture. Any picture parameter set NAL unit containing the value of pic_parameter_set_id for the active view picture parameter set RBSP for a particular value of VOIdx less than VOIdxMax for a coded picture shall have the same content as that of the active view picture parameter set RBSP for the particular value of VOIdx for this coded picture unless it follows the last VCL NAL unit of this coded picture and precedes the first VCL NAL unit of another coded picture.

An MVC sequence parameter set RBSP includes parameters that can be referred to by one or more picture parameter set RBSPs or one or more buffering period SEI messages.

Each MVC sequence parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one MVC sequence parameter set RBSP is considered as the active MVC sequence parameter set RBSP at any given moment during the operation of the decoding process, and when any particular MVC sequence parameter set RBSP becomes the active MVC sequence parameter set RBSP, the previously-active MVC sequence parameter set RBSP (if any) is deactivated.

In addition to the active MVC sequence parameter set RBSP, zero or more MVC sequence parameter set RBSPs may be specifically active for view components (with a particular value of VOIdx less than VOIdxMax) that may be referred to through inter-view prediction in decoding the view component with VOIdx equal to VOIdxMax. Such an MVC sequence parameter set RBSP is referred to as the active view MVC sequence parameter set RBSP for the particular value of VOIdx. The restrictions on active MVC sequence parameter set RBSPs also apply to active view MVC sequence parameter set RBSPs for a particular value of VOIdx less than VOIdxMax.

For the following specification, the activating buffering period SEI message is specified as follows.

– If VOIdxMax is equal to VOIdxMin and the access unit contains a buffering period SEI message not included in an MVC scalable nesting SEI message, this buffering period SEI message is the activating buffering period SEI message.

– Otherwise if VOIdxMax is not equal to VOIdxMin and the access unit contains a buffering period SEI message included in an MVC scalable nesting SEI message and associated with the operation point being decoded, this buffering period SEI message is the activating buffering period SEI message.

– Otherwise, the access unit does not contain an activating buffering period SEI message.

When a sequence parameter set RBSP (nal_unit_type is equal to 7) with a particular value of seq_parameter_set_id is not already the active MVC sequence parameter set RBSP and it is referred to by activation of a picture parameter set RBSP (using that value of seq_parameter_set_id) and the picture parameter set RBSP is activated by a coded slice NAL unit with nal_unit_type equal to 1 or 5 (the picture parameter set RBSP becomes the active picture parameter set RBSP and VOIdxMax is equal to VOIdxMin) and the access unit does not contain an activating buffering period SEI message, it is activated. This sequence parameter set RBSP is called the active MVC sequence parameter set RBSP until it is deactivated when another MVC sequence parameter set RBSP becomes the active MVC sequence parameter set RBSP. A sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation.

When a sequence parameter set RBSP (nal_unit_type is equal to 7) with a particular value of seq_parameter_set_id is not already the active MVC sequence parameter set RBSP and it is referred to by an activating buffering period SEI message (using that value of seq_parameter_set_id) that is not included in an MVC scalable nesting SEI message and VOIdxMax is equal to VOIdxMin, it is activated. This sequence parameter set RBSP is called the active MVC sequence parameter set RBSP until it is deactivated when another MVC sequence parameter set RBSP becomes the active MVC sequence parameter set RBSP. A sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation.

When a subset sequence parameter set RBSP (nal_unit_type is equal to 15) with a particular value of seq_parameter_set_id is not already the active MVC sequence parameter set RBSP and it is referred to by activation of a picture parameter set RBSP (using that value of seq_parameter_set_id) and the picture parameter set RBSP is activated by a coded slice extension NAL unit (nal_unit_type is equal to 20) with VOIdx equal to VOIdxMax (the picture parameter set RBSP becomes the active picture parameter set RBSP) and the access unit does not contain an activating buffering period SEI message, it is activated. This subset sequence parameter set RBSP is called the active MVC sequence parameter set RBSP until it is deactivated when another MVC sequence parameter set RBSP becomes the active MVC sequence parameter set RBSP. A subset sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation.

When a subset sequence parameter set RBSP (nal_unit_type is equal to 15) with a particular value of seq_parameter_set_id is not already the active MVC sequence parameter set RBSP and it is referred to by an activating buffering period SEI message (using that value of seq_parameter_set_id) that is included in an MVC scalable nesting SEI message, it is activated. This subset sequence parameter set RBSP is called the active MVC sequence parameter set RBSP until it is deactivated when another MVC sequence parameter set RBSP becomes the active MVC sequence parameter set RBSP. A subset sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation.

NOTE 2 – The active MVC sequence parameter set RBSP is either a sequence parameter set RBSP or a subset sequence parameter set RBSP. Sequence parameter set RBSPs are activated by coded slice NAL units with nal_unit_type equal to 1 or 5 or buffering period SEI messages that are not included in an MVC scalable nesting SEI message. Subset sequence parameter sets are activated by coded slice extension NAL units (nal_unit_type equal to 20) or buffering period SEI messages that are included in an MVC scalable nesting SEI message. A sequence parameter set RBSP and a subset sequence parameter set RBSP may have the same value of seq_parameter_set_id.

For the following specification, the activating view buffering period SEI message for a particular value of VOIdx is specified as follows.

– If the access unit contains one or more than one buffering period SEI message included in an MVC scalable nesting SEI message and associated with an operation point for which the greatest VOIdx in the associated bitstream sbuset is equal to the particular value of VOIdx, the first of these buffering period SEI messages, in decoding order, is the activating view buffering period SEI message for the particular value of VOIdx.

– Otherwise, if the access unit contains a buffering period SEI message not included in an MVC scalable nesting SEI message, this buffering period SEI message is the activating view buffering period SEI message for the particular value of VOIdx equal to VOIdxMin.

– Otherwise, the access unit does not contain an activating buffering period SEI message for the particular value of VOIdx.

When a sequence parameter set RBSP (nal_unit_type is equal to 7) with a particular value of seq_parameter_set_id is not already the active view MVC sequence parameter set RBSP for VOIdx equal to VOIdxMin and VOIdxMax is greater than VOIdxMin and it is referred to by activation of a picture parameter set RBSP (using that value of seq_parameter_set_id) and the picture parameter set RBSP is activated by a coded slice NAL unit with nal_unit_type equal to 1 or 5 (the picture parameter set RBSP becomes the active view picture parameter set RBSP for VOIdx equal to

VOIdxMin), it is activated for view components with VOIdx equal to VOIdxMin. This sequence parameter set RBSP is called the active view MVC sequence parameter set RBSP for VOIdx equal to VOIdxMin until it is deactivated when another MVC sequence parameter set RBSP becomes the active view MVC sequence parameter set RBSP for VOIdx equal to VOIdxMin or when decoding an access unit with VOIdxMax equal to VOIdxMin, whichever is earlier. A sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation.

When a sequence parameter set RBSP (nal_unit_type is equal to 7) with a particular value of seq_parameter_set_id is not already the active view MVC sequence parameter set RBSP for VOIdx equal to VOIdxMin and VOIdxMax is greater than VOIdxMin and it is referred to by an activating view buffering period SEI message (using that value of seq_parameter_set_id) that is not included in an MVC scalable nesting SEI message, the sequence parameter set RBSP is activated for view components with VOIdx equal to VOIdxMin. This sequence parameter set RBSP is called the active view MVC sequence parameter set RBSP for VOIdx equal to VOIdxMin until it is deactivated when another MVC sequence parameter set RBSP becomes the active view MVC sequence parameter set RBSP for VOIdx equal to VOIdxMin or when decoding an access unit with VOIdxMax equal to VOIdxMin. A sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation.

When a subset sequence parameter set RBSP (nal_unit_type is equal to 15) with a particular value of seq_parameter_set_id is not already the active view MVC sequence parameter set RBSP for a particular value of VOIdx less than VOIdxMax and it is referred to by activation of a picture parameter set RBSP (using that value of seq_parameter_set_id) and the picture parameter set RBSP is activated by a coded slice extension NAL unit (nal_unit_type equal to 20) with the particular value of VOIdx (the picture parameter set RBSP becomes the active view picture parameter set RBSP for the particular value of VOIdx), it is activated for view components with the particular value of VOIdx. This subset sequence parameter set RBSP is called the active view MVC sequence parameter set RBSP for the particular value of VOIdx until it is deactivated when another MVC sequence parameter set RBSP becomes the active view MVC sequence parameter set RBSP for the particular value of VOIdx or when decoding an access unit with VOIdxMax less than or equal to the particular value of VOIdx. A subset sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation.

When a subset sequence parameter set RBSP (nal_unit_type is equal to 15) with a particular value of seq_parameter_set_id is not already the active view MVC sequence parameter set RBSP for a particular value of VOIdx less than VOIdxMax and it is referred to by an activating view buffering period SEI message (using that value of seq_parameter_set_id) that is included in an MVC scalable nesting SEI message and associated with the particular value of VOIdx, this subset sequence parameter set RBSP is activated for view components with the particular value of VOIdx. This subset sequence parameter set RBSP is called the active view MVC sequence parameter set RBSP for the particular value of VOIdx until it is deactivated when another MVC sequence parameter set RBSP becomes the active view MVC sequence parameter set RBSP for the particular value of VOIdx or when decoding an access unit with VOIdxMax less than or equal to the particular value of VOIdx. A subset sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation.

An MVC sequence parameter set RBSP that includes a value of profile_idc not specified in Annex A or Annex H shall not be referred to by activation of a picture parameter set RBSP as the active picture parameter set RBSP or as active view picture parameter set RBSP (using that value of seq_parameter_set_id) or referred to by a buffering period SEI message (using that value of seq_parameter_set_id). An MVC sequence parameter set RBSP including a value of profile_idc not specified in Annex A or Annex H is ignored in the decoding for profiles specified in Annex A or Annex H.

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

– For each particular value of VOIdx, all coded slice NAL units of a coded video sequence shall refer to the same value of seq_parameter_set_id (via the picture parameter set RBSP that is referred to by the value of pic_parameter_set_id).

– The value of seq_parameter_set_id in a buffering period SEI message that is not included in an MVC scalable nesting SEI message shall be identical to the value of seq_parameter_set_id in the picture parameter set RBSP that is referred to by coded slice NAL units (with nal_unit_type equal to 1 or 5) (via the value of pic_parameter_set_id) in the same access unit.

– The value of seq_parameter_set_id in a buffering period SEI message that is included in an MVC scalable nesting SEI message and is associated with a particular value of VOIdx shall be identical to the value of seq_parameter_set_id in the picture parameter set RBSP that is referred to by coded slice NAL units with the particular value of VOIdx (via the value of pic_parameter_set_id) in the same access unit.

The active view MVC sequence parameter set RBSPs for different values of VOIdx may be the same MVC sequence parameter set RBSP. The active MVC sequence parameter set RBSP and an active view MVC sequence parameter set RBSP for a particular value of VOIdx may be the same MVC sequence parameter set RBSP.

When the active MVC sequence parameter set RBSP for a coded picture is a sequence parameter set RBSP, any sequence parameter set RBSP in the coded video sequence containing this coded picture and with the value of seq_parameter_set_id for the active MVC sequence parameter set RBSP shall have the same content as that of the active MVC sequence parameter set RBSP.

When the active MVC sequence parameter set RBSP for a coded picture is a subset sequence parameter set RBSP, any subset sequence parameter set RBSP in the coded video sequence containing this coded picture and with the value of seq_parameter_set_id for the active MVC sequence parameter set RBSP shall have the same content as that of the active MVC sequence parameter set RBSP.

For each particular value of VOIdx, the following applies:

– When the active view MVC sequence parameter set RBSP for a coded picture is a sequence parameter set RBSP, any sequence parameter set RBSP in the coded video sequence containing this coded picture and with the value of seq_parameter_set_id for the active view MVC sequence parameter set RBSP shall have the same content as that of the active view MVC sequence parameter set RBSP.

– When the active view MVC sequence parameter set RBSP for a coded picture is a subset sequence parameter set RBSP, any subset sequence parameter set RBSP in the coded video sequence containing this coded picture and with the value of seq_parameter_set_id for the active view MVC sequence parameter set RBSP shall have the same content as that of the active view MVC sequence parameter set RBSP.

NOTE 3 – If picture parameter set RBSPs or MVC sequence parameter set RBSPs are conveyed within the bitstream, these constraints impose an order constraint on the NAL units that contain the picture parameter set RBSPs or MVC sequence parameter set RBSPs, respectively. Otherwise (picture parameter set RBSPs or MVC sequence parameter set RBSPs are conveyed by other means not specified in this Recommendation | International Standard), they must be available to the decoding process in a timely fashion such that these constraints are obeyed.

When present, a sequence parameter set extension RBSP includes parameters having a similar function to those of a sequence parameter set RBSP. For purposes of establishing constraints on the syntax elements of the sequence parameter set extension RBSP and for purposes of determining activation of a sequence parameter set extension RBSP, the sequence parameter set extension RBSP shall be considered part of the preceding sequence parameter set RBSP with the same value of seq_parameter_set_id. When a sequence parameter set RBSP is present that is not followed by a sequence parameter set extension RBSP with the same value of seq_parameter_set_id prior to the activation of the sequence parameter set RBSP, the sequence parameter set extension RBSP and its syntax elements shall be considered not present for the active sequence parameter set RBSP. The contents of sequence parameter set extension RBSPs only apply when the base view, which conforms to one or more of the profiles specified in Annex A, of a coded video sequence conforming to one or more profiles specified in Annex H is decoded. Subset sequence parameter set RBSPs shall not be followed by a sequence parameter set extension RBSP.

NOTE 4 – Sequence parameter sets extension RBSPs are not considered to be part of a subset sequence parameter set RBSP and subset sequence parameter set RBSPs must not be followed by a sequence parameter set extension RBSP.

For view components with VOIdx equal to VOIdxMax, all constraints that are expressed on the relationship between the values of the syntax elements (and the values of variables derived from those syntax elements) in MVC sequence parameter sets and picture parameter sets and other syntax elements are expressions of constraints that apply only to the active MVC sequence parameter set and the active picture parameter set. For view components with a particular value of VOIdx less than VOIdxMax, all constraints that are expressed on the relationship between the values of the syntax elements (and the values of variables derived from those syntax elements) in MVC sequence parameter sets and picture parameter sets and other syntax elements are expressions of constraints that apply only to the active view MVC sequence parameter set and the active view picture parameter set for the particular value of VOIdx. If any MVC sequence parameter set RBSP having profile_idc equal to one of the profile_idc values specified in Annex A or Annex H is present that is never activated in the bitstream (i.e., it never becomes the active MVC sequence parameter set or an active view MVC sequence parameter set), its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise-conforming bitstream. If any picture parameter set RBSP is present that is never activated in the bitstream (i.e., it never becomes the active picture parameter set or an active view picture parameter set), its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise-conforming bitstream.

During operation of the decoding process (see subclause H.8), for view components with VOIdx equal to VOIdxMax, the values of parameters of the active picture parameter set and the active MVC sequence parameter set shall be considered

in effect. For view components with a particular value of VOIdx less than VOIdxMax, the values of the parameters of the active view picture parameter set and the active view MVC sequence parameter set for the particular value of VOIdx shall be considered in effect. For interpretation of SEI messages that apply to the entire access unit or the view component with VOIdx equal to VOIdxMax, the values of the parameters of the active picture parameter set and the active MVC sequence parameter set for the same access unit shall be considered in effect unless otherwise specified in the SEI message semantics. For interpretation of SEI messages that apply to view components with a particular value of VOIdx less than VOIdxMax, the values of the parameters of the active view picture parameter set and the active view MVC sequence parameter set for the particular value of VOIdx for the same access unit shall be considered in effect unless otherwise specified in the SEI message semantics.

H.7.4.1.2.2 Order of access units and association to coded video sequences

The specification of subclause 7.4.1.2.2 applies with the following modifications.

The first access unit of the bitstream shall only contain coded slice NAL units with nal_unit_type equal to 5 or idr_flag equal to 1.

The order of NAL units and coded pictures and their association to access units is described in subclause H.7.4.1.2.3.

H.7.4.1.2.3 Order of NAL units and coded pictures and association to access units

The specification of subclause 7.4.1.2.3 applies with the following modifications.

The association of VCL NAL units to primary or redundant coded pictures is specified in subclause H.7.4.1.2.5.

The constraints for the detection of the first VCL NAL unit of a primary coded picture are specified in subclause H.7.4.1.2.4.

The constraint expressed in subclause 7.4.1.2.3 on the order of a buffering period SEI message is replaced by the following constraints.

- When an SEI NAL unit containing a buffering period SEI message is present, the following applies:
  - If the buffering period SEI message is the only buffering period SEI message in the access unit and it is not included in an MVC scalable nesting SEI message, the buffering period SEI message shall be the first SEI message payload of the first SEI NAL unit in the access unit.
  - Otherwise (the buffering period SEI message is not the only buffering period SEI message in the access unit or it is included in an MVC scalable nesting SEI message), the following applies:
    - When a buffering period SEI message that is not included in an MVC scalable nesting SEI message is present, this buffering period SEI message shall be the only SEI message payload of the first SEI NAL unit in the access unit.
    - An MVC scalable nesting SEI message that includes a buffering period SEI message shall not include any other SEI messages and shall be the only SEI message inside the SEI NAL unit.
    - All SEI NAL units that precede an SEI NAL unit that contains an MVC scalable nesting SEI message with a buffering period SEI message as payload in an access unit shall only contain buffering period SEI messages or MVC scalable nesting SEI messages with a buffering period SEI message as payload.

The following additional constraints shall be obeyed.

- Each NAL unit with nal_unit_type equal to 1 or 5 shall be immediately preceded by a prefix NAL unit.
- Each prefix NAL unit shall be immediately followed by a NAL unit with nal_unit_type equal to 1 or 5.

H.7.4.1.2.4 Detection of the first VCL NAL unit of a primary coded picture

This subclause specifies constraints on VCL NAL unit syntax that are sufficient to enable the detection of the first VCL NAL unit of each primary coded picture.

The first VCL NAL unit of the primary coded picture of the current access unit, in decoding order, shall be different from the last VCL NAL unit of the primary coded picture of the previous access unit, in decoding order, in one or more of the following ways.

- view_id of the first VCL NAL unit of the primary coded picture of the current access unit is differnt from view_id of the last VCL NAL unit of the primary coded picture of the previous access unit, and VOIdx of the first VCL NAL

unit of the primary coded picture of the current access unit is smaller than VOIdx of the last VCL NAL unit of the primary coded picture of the previous access unit

– view_id of the first VCL NAL unit of the primary coded picture of the current access unit and the last VCL NAL unit of the primary coded picture of the previous access unit is identical, and any of the conditions specified in subclause 7.4.1.2.4 is fulfilled

H.7.4.1.2.5 Order of VCL NAL units and association to coded pictures

Each VCL NAL unit is part of a coded picture.

Let voIdx be the value of VOIdx of any particular VCL NAL unit. The order of the VCL NAL units within a coded picture is constrained as follows.

– For all VCL NAL units following this particular VCL NAL unit, the value of VOIdx shall be greater than or equal to voIdx.

For each set of VCL NAL units within a view component, the following applies:

– If arbitrary slice order, as specified in Annex A or subclause H.10, is allowed, coded slice NAL units of a view component may have any order relative to each other.

– Otherwise (arbitrary slice order is not allowed), coded slice NAL units of a slice group shall not be interleaved with coded slice NAL units of another slice group and the order of coded slice NAL units within a slice group shall be in the order of increasing macroblock address for the first macroblock of each coded slice NAL unit of the same slice group.

NAL units having nal_unit_type equal to 12 may be present in the access unit but shall not precede the first VCL NAL unit of the primary coded picture within the access unit.

NAL units having nal_unit_type equal to 0 or in the range of 24 to 31, inclusive, which are unspecified, may be present in the access unit but shall not precede the first VCL NAL unit of the primary coded picture within the access unit.

NAL units having nal_unit_type in the range of 21 to 23, inclusive, which are reserved, shall not precede the first VCL NAL unit of the primary coded picture within the access unit (when specified in the future by ITU-T | ISO/IEC).

H.7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics

H.7.4.2.1 Sequence parameter set RBSP semantics

The semantics specified in subclause 7.4.2.1 apply.

H.7.4.2.1.1 Sequence parameter set data semantics

The semantics specified in subclause 7.4.2.1.1 apply with the substitution of MVC sequence parameter set for sequence parameter set. All constraints specified in subclause 7.4.2.1.1 apply only to the view components for which the MVC sequence parameter set is the active MVC sequence parameter set or the active view MVC sequence parameter set as specified in subclause H.7.4.1.2.1.

Additionally, the following changed semantics of max_num_ref_frame apply.

If included in a sequence parameter set, the semantics specified in subclause 7.4.2.1.1 apply. Otherwise (included in a subset sequence parameter set), the following applies.

max_num_ref_frames specifies the maximum number of short-term and long-term reference frames, complementary reference field pairs, and non-paired reference fields that may be used by the decoding process for inter prediction of any view component in the coded video sequence. max_num_ref_frames also determines the sliding window size of the sliding window operation as specified in subclause H.8.3. The value of max_num_ref_frames shall be in the range of 0 to 16, inclusive.

H.7.4.2.1.1.1 Scaling list semantics

The semantics specified in subclause 7.4.2.1.1.1 apply.

H.7.4.2.1.2 Sequence parameter set extension RBSP semantics

The semantics specified in subclause 7.4.2.1.2 apply. Additionally, the following applies.

Sequence parameter set extension RBSPs can only follow sequence parameter set RBSPs in decoding order. Subset sequence parameter set RBSPs shall not be followed by a sequence parameter set extension RBSP. The contents of sequence parameter set extension RBSPs only apply when the base view, which conforms to one or more of the profiles specified in Annex A, of a coded video sequence conforming to one or more profiles specified in Annex H is decoded.

H.7.4.2.1.3 Subset sequence parameter set RBSP semantics

bit_equal_to_one shall be equal to 1.

NOTE 1 – This bit enables the syntax structure of subset_seq_parameter_set_rbsp( ) to be compatible with the corresponding syntax specified in Annex G.

mvc_vui_parameters_present_flag equal to 0 indicates that the syntax structure mvc_vui_parameters_extension( ) is not present. mvc_vui_parameters_present_flag equal to 1 indicates that the syntax structure mvc_vui_parameters_extension( ) is present.

additional_extension2_flag shall be equal to 0. The value of 1 for additional_extension2_flag is reserved for future use by ITU-T | ISO/IEC.

additional_extension2_data_flag may have any value. It shall not affect the conformance to profiles specified in Annex A, Annex G, or Annex H.

H.7.4.2.1.4 Sequence parameter set MVC extension semantics

The sequence parameter set MVC extension specifies inter-view dependency relationships for the coded video sequence. The sequence parameter set MVC extension also specifies level values for a subset of the operation points for the coded video sequence. All sequence parameter set MVC extensions that are referred to by a coded video sequence shall be identical.

Some views identified by view_id[i] may be not present in the coded video sequence.

NOTE 1 – Some views or temporal subsets described by the sequence parameter set MVC extension may have been removed from the original coded video sequence, hence may be not present in the coded video sequence. However, the information in the sequence parameter set MVC extension always applies to the remaining views and temporal subsets.

num_views_minus1 plus 1 specifies the maximum number of coded views in the coded video sequence. The value of num_view_minus1 shall be in the range of 0 to 1023, inclusive.

NOTE 2 – The actual number of views in the coded video sequence may be less than num_views_minus1 plus 1.

view_id[i **]** specifies the view_id of the view with VOIdx equal to i.

num_anchor_refs_l0[i **]** specifies the number of view components for inter-view prediction in the initialised RefPicList0 in decoding anchor view components with VOIdx equal to i. The value of num_anchor_refs_l0[i] shall not be greater than 15. The value of num_anchor_refs_l0[0] shall be equal to 0.

anchor_ref_l0[i **][** j **]** specifies the view_id of the j-th view component for inter-view prediction in the initialised RefPicList0 in decoding anchor view components with VOIdx equal to i.

num_anchor_refs_l1[i **]** specifies the number of view components for inter-view prediction gin the initialised RefPicList1 in decoding anchor view components with VOIdx equal to i. The value of num_anchor_refsl1[i] shall not be greater than 15. The value of num_anchor_refs_l1[0] shall be equal to 0.

anchor_ref_l1[i **][** j **]** specifies the view_id of the j-th view component for inter-view prediction in the initialised RefPicList1 in decoding an anchor view component with VOIdx equal to i.

num_non_anchor_refs_l0[i **]** specifies the number of view components for inter-view prediction in the initialised RefPicList0 in decoding non-anchor view components with VOIdx equal to i. The value of num_non_anchor_refsl0[i] shall not be greater than 15. The value of num_non_anchor_refs_l0[0] shall be equal to 0.

non_anchor_ref_l0[i **][** j **]** specifies the view_id of the j-th view component for inter-view prediction in the initialised RefPicList0 in decoding non-anchor view components with VOIdx equal to i.

num_non_anchor_refs_l1[i **]** specifies the number of view components for inter-view prediction in the initialised RefPicList1 in decoding non-anchor view components with VOIdx equal to i. The value of num_non_anchor_refsl1[i] shall not be greater than 15. The value of num_non_anchor_refs_l1[0] shall be equal to 0.

non_anchor_ref_l1[i **][** j **]** specfies the view_id of the j-th view component for inter-view prediction in the initialised RefPicList1 in decoding non-anchor view components with VOIdx equal to i.

For any particular view with view_id equal to vId1 and VOIdx equal to vOIdx1 and another view with view_id equal to vId2 and VOIdx equal to vOIdx2, when vId2 is equal to one of non_anchor_ref_l0[vOIdx1][j] for all j in the range of 0 to num_non_anchor_refs_l0[vOIdx1], exclusive, or one of non_anchor_ref_l1[vOIdx1][j] for all j in the range of 0 to num_non_anchor_refs_l1[vOIdx1], exclusive, vId2 shall also be equal to one of anchor_ref_l0[vOIdx1][j] for all j in the range of 0 to num_anchor_refs_l0[vOIdx1], exclusive, or one of anchor_ref_l1[vOIdx1][j] for all j in the range of 0 to num_anchor_refs_l1[vOIdx1], exclusive.

NOTE 3 – The inter-view dependency for non-anchor view components is a subset of that for anchor view components.

num_level_values_signalled specfies the number of level values signalled for the coded video sequence. The value of num_level_values_signalled shall be in the range of 0 to 63, inclusive.

level_idc[i **]** specifies the i-th level value signalled for the coded video sequence.

num_applicable_ops_minus1[i **]** plus 1 specifies the number of operation points to which the level indicated by level_idc1[i] applies. The value of num_applicable_ops_minus1[i] shall be in the range of 0 to 1023, inclusive.

applicable_op_temporal_id[i **][** j **]** specifies the temporal_id of the j-th operation point to which the the level indicated by level_idc[i] applies.

applicable_op_num_target_views_minus1[i **][** j **]** specifies the number of target output views for the j-th operation point to which the the level indicated by level_idc[i] applies. The value of applicable_op_num_target_views_minus1[i][j] shall be in the range of 0 to 1023, inclusive.

applicable_op_target_view_id[i **][** j **][** k **]** specifies the k-th target output view for the j-th operation point to which the the level indicated by level_idc[i] applies. The value of applicable_op_num_target_views_minus1[i][j] shall be in the range of 0 to 1023, inclusive.

Let maxTId be the greatest temporal_id of all NAL units in the coded video sequence, and vId be view_id of any view in the coded video sequence. There shall be one set of applicable_op_temporal_id[i][j], applicable_op_num_target_views_minus1[i][j], and applicable_op_target_view_id[i][j][k], for any i and j and all k for the i and j, in which applicable_op_temporal_id[i][j] is equal to maxTId, applicable_op_num_target_views_minus1[i][j] is equal to 0, and applicable_op_target_view_id[i][j][k] is equal to vId.

NOTE 4 – The above constraint ensures that the level applying to each operation point consisting of only one target output view with the greatest highest temporal_id in the coded video sequence is signalled by one of the level_idc[i] for all i.

NOTE 5 – Some operation points identified by applicable_op_temporal_id[i][j], applicable_op_num_target_views_minus1[i][j], and applicable_op_target_view_id[i][j][k], for all i, j, and k, may be not present in the coded video sequence.

applicable_op_num_views_minus1[i **][** j **]** specifies the number of views, including the views that are dependent on by the target output views but do not belong to the target output views, in the j-th operation point to which the the level indicated by level_idc[i] applies. The value of applicable_op_num_views_minus1[i][j] shall be in the range of 0 to 1023, inclusive.

H.7.4.2.2 Picture parameter set RBSP semantics

The semantics specified in subclause 7.4.2.2 apply with substituting MVC sequence parameter set for sequence parameter set. All constraints specified in subclause 7.4.2.2 apply only to the view components for which the picture parameter set is the active picture parameter set or the active view picture parameter set as specified in subclause H.7.4.1.2.1.

H.7.4.2.3 Supplemental enhancement information RBSP semantics

The semantics specified in subclause 7.4.2.3 apply.

H.7.4.2.3.1 Supplemental enhancement information message semantics

The semantics specified in subclause 7.4.2.3.1 apply.

H.7.4.2.4 Access unit delimiter RBSP semantics

The semantics specified in subclause 7.4.2.4 apply.

H.7.4.2.5 End of sequence RBSP semantics

The semantics specified in subclause 7.4.2.5 apply.

H.7.4.2.6 End of stream RBSP semantics

The semantics specified in subclause 7.4.2.6 apply.

H.7.4.2.7 Filler data RBSP semantics

The semantics specified in subclause 7.4.2.7 apply with the following addition.

Filler data NAL units shall be considered to contain the syntax elements priority_id, view_id, and temporal_id with values that are inferred as follows.

– Let prevMvcNalUnit be the most recent NAL unit in decoding order that has nal_unit_type equal to 14 or 20.

NOTE – The most recent NAL unit in decoding order with nal_unit_type equal to 14 or 20 always belongs to the same access unit as the filler data NAL unit.

– The values of priority_id, view_id, and temporal_id for the filler data NAL unit are inferred to be equal to the values of priority_id, view_id, and temporal_id, respectively, of the NAL unit prevMvcNalUnit.

H.7.4.2.8 Slice layer without partitioning RBSP semantics

The semantics specified in subclause 7.4.2.8 apply.

H.7.4.2.9 Slice data partition RBSP semantics

Slice data partition syntax is not present in bitstreams conforming to one or more of the profiles specified in Annex H.

H.7.4.2.10 RBSP slice trailing bits semantics

The semantics specified in subclause 7.4.2.10 apply.

H.7.4.2.11 RBSP trailing bits semantics

The semantics specified in subclause 7.4.2.11 apply.

H.7.4.2.12 Prefix NAL unit RBSP semantics

The prefix NAL unit RBSP consists of the one-byte NAL unit header as specified in subclauses 7.3.1 and 7.4.1, and the three-byte NAL unit header extension as specified in subclauses H.7.3.1.1 and H.7.4.1.1.

H.7.4.2.13 Slice layer extension RBSP semantics

The slice layer extension RBSP consists of a slice header and slice data.

H.7.4.3 Slice header semantics

The semantics specified in subclause 7.4.3 apply with the following modifications.

All constraints specified in subclause 7.4.3 apply only to the view components with the same value of VOIdx.

The value of the following MVC sequence parameter set syntax elements shall be the same across all coded slice NAL units of an access unit: chroma_format_idc.

frame_num is used as an identifier for view components and shall be represented by log2_max_frame_num_minus4 + 4 bits in the bitstream.

frame_num is constrained as specified in subclause 7.4.3 where this constraint applies to view components with view_id equal to the current value of view_id.

direct_spatial_mv_pred_flag is as specified in subclause 7.4.3 with the following modification.

When RefPicList1[0] points to an inter-view reference picture or an inter-view only reference component, which belongs to the same access unit as the current view component, direct_spatial_mv_pred_flag shall be equal to 1.

weighted_bipred_idc is as specified in subclause 7.4.3 with the folloiwng modification.

When there is at least one inter-view prediction reference, which belongs to the same access unit as the current view component, in RefPicList0 or RefPicList1, weighted_bipred_idc shall not be equal to 2.

H.7.4.3.1 Reference picture list modification semantics

The semantic specified in subclauses 7.4.3.1 applies with the following modified semantics of modification_of_pic_nums_idc. In addition, the semantics of abs_diff_view_idx_minus1 specified below applies.

modification_of_pic_nums_idc together with abs_diff_pic_num_minus1, long_term_pic_num, or abs_diff_view_idx_minus1 specifies which of the reference pictures or inter-view only reference components are re-mapped. The values of modification_of_pic_nums_idc are specified in Table H-1 The value of the first modification_of_pic_nums_idc that follows immediately after ref_pic_list_modification_flag_l0 or ref_pic_list_modification_flag_l1 shall not be equal to 3.

Table H-1 – modification_of_pic_nums_idc operations for modification of reference picture lists

modification_of_pic_nums_idc	Modification specified
0	abs_diff_pic_num_minus1 is present and corresponds to a difference to subtract from a picture number prediction value
1	abs_diff_pic_num_minus1 is present and corresponds to a difference to add to a picture number prediction value
2	long_term_pic_num is present and specifies the long-term picture number for a reference picture
3	End loop for modification of the initial reference picture list
4	abs_diff_view_idx_minus1 is present and corresponds to a difference to subtract from a prediction value of the inter-view reference index
5	abs_diff_view_idx_minus1 is present and corresponds to a difference to add to a prediction value of the inter-view reference index

abs_diff_view_idx_minus1 plus 1 specifies the absolute difference between the inter-view reference index to put to the current index in the reference picture list and the prediction value of the inter-view reference index.

Let currVOIdx be the VOIdx of the current view component, and let intViewIdx be the inter-view reference index of the target inter-view prediction reference to put to the current index in RefPicListX (X is 0 or 1).

– If the current view component is an anchor view component, the view_id of the target inter-view prediction reference is equal to anchor_ref_lX[currVOIdx][intViewIdx]. For anchor view components with VOIdx equal to currVOIdx, abs_diff_view_idx_minus1 shall be in the range of 0 to max(0, num_anchor_refs_lX[currVOIdx] − 1), inclusive.

– Otherwise (the current view component is not an anchor view component), the view_id of the target inter-view prediction reference is equal to non_anchor_ref_lX[currVOIdx] [intViewIdx]. For non-anchor view components with VOIdx equal to currVOIdx, abs_diff_view_idx_minus1 shall be in the range of 0 to max(0, num_non_anchor_refs_lX[currVOIdx] − 1), inclusive.

The allowed values of abs_diff_view_idx_minus1 are further restricted as specified in subclause H.8.2.2.3.

H.7.4.3.2 Prediction weight table semantics

The semantics specified in subclause 7.4.3.2 apply.

H.7.4.3.3 Decoded reference picture marking semantics

The semantics specified in subclause 7.4.3.3 apply to each view independently, with "sequence parameter set" being replaced by "MVC sequence parameter set", and "primary coded picture" being replaced by "primary view component".

H.7.4.4 Slice data semantics

The semantics specified in subclause 7.4.4 apply.

H.7.4.5 Macroblock layer semantics

The semantics specified in subclause 7.4.5 apply.

H.7.4.5.1 Macroblock prediction semantics

The semantics specified in subclause 7.4.5.1 apply.

H.7.4.5.2 Sub-macroblock prediction semantics

The semantics specified in subclause 7.4.5.2 apply.

H.7.4.5.3 Residual data semantics

The semantics specified in subclause 7.4.5.3 apply.

H.7.4.5.3.1 Residual luma semantics

The semantics specified in subclause 7.4.5.3.1 apply.

H.7.4.5.3.2 Residual block CAVLC semantics

The semantics specified in subclause 7.4.5.3.2 apply.

H.7.4.5.3.3 Residual block CABAC semantics

The semantics specified in subclause 7.4.5.3.3 apply.

H.8 MVC decoding process

This subclause specifies the decoding process for an access unit of a coded video sequence conforming to one or more of the profiles specified in Annex H. Specifically, this subclause specifies how the decoded picture with multiple view components is derived from syntax elements and global variables that are derived from NAL units in an access unit when the decoder is decoding the operation point identified by the target temporal level and the target output views.

The decoding process is specified such that all decoders shall produce numerically identical results for the target output views. Any decoding process that produces identical results for the target output views to the process described here conforms to the decoding process requirements of this Recommendation | International Standard.

Unless stated otherwise, the syntax elements and derived upper-case variables that are referred to by the decoding process specified in this subclause and all child processes invoked from the process specified in this subclause are the syntax elements and derived upper-case variables for the current access unit.

The target output views are either specified by external means not specified in this Specification, or, when not specified by external means, there shall be one target output view which is the base view. Let OutputVOIdxList be the list of VOIdx values, in increasing order of VOIdx, of all the target output views in one access unit. The list OutputVOIdxList shall not change within a coded video sequence.

All sub-bitstreams that can be derived using the sub-bitstream extraction process with pIdTarget equal to any value in the range of 0 to 63, inclusive, tIdTarget equal to any value in the range of 0 to 7, inclusive, viewIdTargetList consisting of any one or more viewIdTarget's identifying the views in the bitstream as inputs as specified in subclause H.8.4 shall result in a set of coded video sequences, with each coded video sequence conforming to one or more of the profiles specified in Annex A and Annex H.

Let vOIdxList be a list of integer values specifying the VOIdx values of the view components of the access unit. The variable VOIdxMax is set equal to the maximum value of the entries in the list vOIdxList, and the variable vOIdxMin is set to the minimum value of the entries in the list vOIdxList. VOIdxMax shall be the same for all access units within a coded video sequence. vOIdxMin shall be the same for all anchor access units within a coded video sequence. When the current access unit is an anchor access unit, the variable VOIdxMin is set to vOIdxMin.

The multiview video decoding process specified in this subclause is repeatedly invoked for each view component with VOIdx from vOIdxMin to VOIdxMax, inclusive, which is present in the list vOIdxList, in increasing order of VOIdx.

Outputs of the multiview video decoding process are decoded samples of the current primary coded picture including all decoded view components.

The specifications in clause 8 shall apply, with the decoding processes for picture order count, reference picture lists

construction and decoded reference picture marking being modified in subclauses H.8.1, H.8.2 and H.8.3, respectively. Additionally, the specification of bitstream subsets is specified in H.8.4.

H.8.1 MVC decoding process for picture order count

The specifications in subclause 8.2.1 apply independently for each view.

H.8.2 MVC decoding process for reference picture lists construction

This process is invoked at the beginning of the decoding process for each P, SP or B slice.

Duing the invocation of this process, when subclauses 8.2.4.1 and 8.2.4.2 are invoked, only the reference pictures having the same value of view_id as the current slice are considered.

Decoded reference pictures are marked as "used for short-term reference" or "used for long-term reference" as specified in subclause H.8.3. Short-term reference pictures are identified by the values of frame_num and view_id, and, for inter-view reference pictures, additionally by PicOrderCnt(). Long-term reference pictures are assigned a long-term frame index as specified in subclause H.8.3 and identified by the values long-term frame index, view_id, and, for inter-view reference pictures, additionally by PicOrderCnt().

In addition to reference pictures, inter-view only reference components (which are non-reference pictures and not marked by the reference picture marking process) may also be included in a reference picture list. Inter-view only reference components are identified by the value of view_id and by PicOrderCnt().

Subclause 8.2.4.1 is invoked to specify

– the assignment of variables FrameNum, FrameNumWrap, and PicNum to each of the short-term reference pictures, and

– the assignment of variable LongTermPicNum to each of the long-term reference pictures.

Reference pictures and, when present, inter-view only reference components, are addressed through reference indices as specified in subclause 8.2.4.1. A reference index is an index into a reference picture list. When decoding a P or SP slice, there is a single reference picture list RefPicList0. When decoding a B slice, there is a second independent reference picture list RefPicList1 in addition to RefPicList0.

At the beginning of the decoding process for each slice, reference picture list RefPicList0, and for B slices RefPicList1, are derived as specified by the following ordered steps:

1. The initial reference picture list RefPicList0 and for B slices RefPicList1 are derived as specified in subclause 8.2.4.2.
2. Inter-view reference pictures or inter-view only reference components are appended to the initial reference picture list RefPicList0 and for B slices RefPicList1 as specified in subclause H.8.2.1.
3. When ref_pic_list_modification_flag_l0 is equal to 1 or, when decoding a B slice, ref_pic_list_modification_flag_l1 is equal to 1, the reference picture list RefPicList0 and for B slices RefPicList1 are modified as specified in subclause H.8.2.2.

   NOTE – The modification process for reference picture lists specified in subclause H.8.2.2 allows the contents of RefPicList0 and for B slices RefPicList1 to be modified in a flexible fashion. In particular, it is possible for a reference picture that is currently marked "used for reference" to be inserted into RefPicList0 and for B slices RefPicList1 even when the reference picture is not in the initial reference picture list derived as specified in subclauses 8.2.4.2 and H.8.2.1.

The number of entries in the modified reference picture list RefPicList0 is num_ref_idx_l0_active_minus1 + 1, and for B slices the number of entries in the modified reference picture list RefPicList1 is num_ref_idx_l1_active_minus1 + 1. A reference picture or inter-view only reference component may appear at more than one index in the modified reference picture lists RefPicList0 or RefPicList1.

During the invocation of the process specified in subclause H.8.2.1, an inter-view prediction reference appended to RefPicListX (with X being 0 or 1) may not exist. However, an inter-view prediction reference that does not exist shall not be in the modified RefPicListX after the invocation of the process specified in subclause H.8.2.2.

H.8.2.1 Initilisation process for reference picture list for inter-view prediction references

Inputs to this process are a reference picture list RefPicListX (with X being 0 or 1), inter_view_flag and view dependency information that has been decoded from the seq_parameter_set_mvc_extension().

The output of this process is a possibly modified reference picture list RefPicListX, which is still referred to as the intial

reference picture list RefPicListX.

The following procedure shall be conducted to append inter-view reference pictures and inter-view only reference components into the reference picture list. Let i be the VOIdx of the current slice:

– If the current slice has anchor_pic_flag equal to 1, then for each value of inter-view reference index j from 0 to num_anchor_refs_lX[i] - 1, inclusive, in ascending order of j, the inter-view prediction reference with view_id equal to anchor_ref_lX[i][j] from the same access unit as the current slice is appended to RefPicListX.

– Otherwise (the current slice has anchor_pic_flag equal to 0), then for each value of inter-view reference index j from 0 to num_non_anchor_refs_lX[i] - 1, inclusive, in ascending order of j, the inter-view prediction reference with view_id equal to non_anchor_ref_lX[i][j] from the same access unit as the current slice is appended to RefPicListX.

H.8.2.2 Modification process for reference picture lists

Input to this process is reference picture list RefPicList0 and, when decoding a B slice, also reference picture list RefPicList1.

Outputs of this process are a possibly modified reference picture list RefPicList0 and, when decoding a B slice, also a possibly modified reference picture list RefPicList1.

When ref_pic_list_modification_flag_l0 is equal to 1, the following applies:

– Let refIdxL0 be an index into the reference picture list RefPicList0. It is initially set equal to 0.

– The corresponding syntax elements modification_of_pic_nums_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies:

  – If modification_of_pic_nums_idc is equal to 0 or equal to 1, the process specified in subclause H.8.2.2.1 is invoked with RefPicList0 and refIdxL0 given as input, and the output is assigned to RefPicList0 and refIdxL0.

  – Otherwise, if modification_of_pic_nums_idc is equal to 2, the process specified in subclause H.8.2.2.2 is invoked with RefPicList0 and refIdxL0 given as input, and the output is assigned to RefPicList0 and refIdxL0.

  – Otherwise, if modification_of_pic_nums_idc is equal to 4 or equal to 5, the process specified in subclause H.8.2.2.3 is invoked with RefPicList0 and refIdxL0 given as input, and the output is assigned to RefPicList0 and refIdxL0.

  – Otherwise (modification_of_pic_nums_idc is equal to 3), the modification process for reference picture list RefPicList0 is finished.

When ref_pic_list_modification_flag_l1 is equal to 1, the following applies:

– Let refIdxL1 be an index into the reference picture list RefPicList1. It is initially set equal to 0.

– The corresponding syntax elements modification_of_pic_nums_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies:

  – If modification_of_pic_nums_idc is equal to 0 or equal to 1, the process specified in subclause H.8.2.2.1 is invoked with RefPicList1 and refIdxL1 given as input, and the output is assigned to RefPicList1 and refIdxL1.

  – Otherwise, if modification_of_pic_nums_idc is equal to 2, the process specified in subclause H.8.2.2.2 is invoked with RefPicList1 and refIdxL1 given as input, and the output is assigned to RefPicList1 and refIdxL1.

  – Otherwise, if modification_of_pic_nums_idc is equal to 4 or equal to 5, the process specified in subclause H.8.2.2.3 is invoked with RefPicList1 and refIdxL1 given as input, and the output is assigned to RefPicList1 and refIdxL1.

  – Otherwise (modification_of_pic_nums_idc is equal to 3), the modification process for reference picture list RefPicList1 is finished.

H.8.2.2.1 Modification process of reference picture lists for short-term reference pictures for inter prediction

Inputs to this process are an index refIdxLX and and reference picture list RefPicListX (with X being 0 or 1).

Outputs of this process are an incremented index refIdxLX and a modified reference picture list RefPicListX.

The variable picNumLXNoWrap is derived as follows:

– If modification_of_pic_nums_idc is equal to 0,

```
if( picNumLXPred – ( abs_diff_pic_num_minus1 + 1 ) < 0 )
      picNumLXNoWrap = picNumLXPred – ( abs_diff_pic_num_minus1 + 1 ) + MaxPicNum          (H-1)
else
      picNumLXNoWrap = picNumLXPred – ( abs_diff_pic_num_minus1 + 1 )
```

– Otherwise (modification_of_pic_nums_idc is equal to 1),

```
if( picNumLXPred + ( abs_diff_pic_num_minus1 + 1 )  >=  MaxPicNum )
      picNumLXNoWrap = picNumLXPred + ( abs_diff_pic_num_minus1 + 1 ) – MaxPicNum          (H-2)
else
      picNumLXNoWrap = picNumLXPred + ( abs_diff_pic_num_minus1 + 1 )
```

picNumLXPred is the prediction value for the variable picNumLXNoWrap. When the process specified in this subclause is invoked the first time for a slice (that is, for the first occurrence of modification_of_pic_nums_idc equal to 0 or 1 in the ref_pic_list_modification( ) syntax), picNumL0Pred and picNumL1Pred are initially set equal to CurrPicNum. After each assignment of picNumLXNoWrap, the value of picNumLXNoWrap is assigned to picNumLXPred.

The variable picNumLX is derived as follows:

```
if( picNumLXNoWrap > CurrPicNum )
      picNumLX = picNumLXNoWrap – MaxPicNum                                              (H-3)
else
      picNumLX = picNumLXNoWrap
```

picNumLX shall be equal to the PicNum of a reference picture that is marked as "used for short-term reference" and shall not be equal to the PicNum of a short-term reference picture that is marked as "non-existing".

The following procedure is conducted to place the picture with short-term picture number picNumLX into the index position refIdxLX, shift the position of any other remaining pictures to later in the list, and increment the value of refIdxLX.

```
for( cIdx = num_ref_idx_lX_active_minus1 + 1; cIdx > refIdxLX; cIdx-- )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx – 1]
RefPicListX[ refIdxLX++ ] = short-term reference picture with PicNum equal to picNumLX
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_lX_active_minus1 + 1; cIdx++ )                 (H-4)
    if( PicNumF( RefPicListX[ cIdx ] ) != picNumLX | | viewID(RefPicListX[ cIdx ] ) != currViewID )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]
```

In the above, the function viewID(refpic) returns the view_id of the reference picture refpic, the variable currViewID is equal to the view_id of the current slice, and the function PicNumF(RefPicListX[cIdx]) is derived as follows:

– If the reference picture RefPicListX[cIdx] is marked as "used for short-term reference", PicNumF(RefPicListX[cIdx]) is the PicNum of the picture RefPicListX[cIdx].

– Otherwise (the reference picture RefPicListX[cIdx] is not marked as "used for short-term reference"), PicNumF(RefPicListX[cIdx]) is equal to MaxPicNum.

NOTE 1 – The value of picNumLX can never be equal to MaxPicNum.

NOTE 2 – Within this pseudo-code procedure, the length of the list RefPicListX is temporarily made one element longer than the length needed for the final list. After the execution of this procedure, only elements 0 through num_ref_idx_lX_active_minus1 of the list need to be retained.

H.8.2.2.2 Modification process of reference picture lists for long-term reference pictures for inter prediction

Inputs to this process are an index refIdxLX (with X being 0 or 1) and reference picture list RefPicListX.

Outputs of this process are an incremented index refIdxLX and a modified reference picture list RefPicListX.

The following procedure is conducted to place the picture with long-term picture number long_term_pic_num into the index position refIdxLX, shift the position of any other remaining pictures to later in the list, and increment the value of refIdxLX.

```
for( cIdx = num_ref_idx_lX_active_minus1 + 1; cIdx > refIdxLX; cIdx-- )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx – 1]
```



```
RefPicListX[ refIdxLX++ ] = long-term reference picture with LongTermPicNum equal to long_term_pic_num
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_lX_active_minus1 + 1; cIdx++ )                    (H-5)
    if( LongTermPicNumF( RefPicListX[ cIdx ] ) != long_term_pic_num | |
    viewID(RefPicListX[ cIdx ] ) != currViewID )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]
```

In the above, the function viewID(refpic) returns the view_id of the reference picture refpic, the variable currViewID is equal to the view_id of the current slice, and the function LongTermPicNumF(RefPicListX[cIdx]) is derived as follows:

– If the reference picture RefPicListX[cIdx] is marked as "used for long-term reference", LongTermPicNumF(RefPicListX[cIdx]) is the LongTermPicNum of the picture RefPicListX[cIdx].

– Otherwise (the reference picture RefPicListX[cIdx] is not marked as "used for long-term reference"), LongTermPicNumF(RefPicListX[cIdx]) is equal to 2 * (MaxLongTermFrameIdx + 1).

NOTE 1 – The value of long_term_pic_num can never be equal to.2 * (MaxLongTermFrameIdx + 1).

NOTE 2 – Within this pseudo-code procedure, the length of the list RefPicListX is temporarily made one element longer than the length needed for the final list. After the execution of this procedure, only elements 0 through num_ref_idx_lX_active_minus1 of the list need to be retained.

H.8.2.2.3 Modification process for reference picture lists for inter-view prediction references

Inputs to this process are reference picture list RefPicListX (with X being 0 or 1) and an index refIdxLX into this list.

Outputs of this process are a modified reference picture list RefPicListX (with X being 0 or 1) and an incremented index refIdxLX.

The variable picViewIdxLX is derived as follows:

– If modification_of_pic_nums_idc is equal to 4,

```
if( picViewIdxLXPred – ( abs_diff_view_idx_minus1 + 1 ) < 0 )
    picViewIdxLX = picViewIdxLXPred – ( abs_diff_view_idx_minus1 + 1 ) + maxViewIdx          (H-6)
else
    picViewIdxLX = picViewIdxLXPred – ( abs_diff_view_idx_minus1 + 1 )
```

– Otherwise (modification_of_pic_nums_idc is equal to 5),

```
if( picViewIdxLXPred + ( abs_diff_view_idx_minus1 + 1 ) >= MaxViewIdx )
    picViewIdxLX = picViewIdxLXPred + ( abs_diff_view_idx_minus1 + 1 ) – maxViewIdx          (H-7)
else
    picViewIdxLX = picViewIdxLXPred + ( abs_diff_view_idx_minus1 + 1 )
```

where maxViewIdx is derived as follows (currVOIdx is VOIdx of the current slice):

– If the current slice has anchor_pic_flag equal to 1, maxViewIdx is set equal to num_anchor_refs_lX[currVOIdx].

– Otherwise (the current slice has anchor_pic_flag equal to 0), maxViewIdx is set equal to num_non_anchor_refs_lX[currVOIdx].

picViewIdxLXPred is the prediction value for the variable picViewIdxLX. When the process specified in this subclause is invoked the first time for a slice (that is, for the first occurrence of modification_of_pic_nums_idc equal to 4 or 5 in the ref_pic_list_modification() syntax), picViewIdxL0Pred and picViewIdxL1Pred are initially set equal to 0. After each assignment of picViewIdxLX, the value of picViewIdxLX is assigned to picViewIdxLXPred.

The following procedure is conducted to place the inter-view prediction reference with inter-view reference index equal to picViewIdxLX into the index position refIdxLX and shift the position of any other remaining pictures to later in the list.

– If the current slice has anchor_pic_flag equal to 1,

```
targetViewID = anchor_refs_lX[ currVOIdx ][ picViewIdxLX ]
```

– Otherwise (the current slice has anchor_pic_flag equal to 0),

```
targetViewID = non_anchor_refs_lX[ currVOIdx ][ picViewIdxLX ]
```

```
for( cIdx = num_ref_idx_lX_active_minus1 + 1; cIdx > refIdxLX; cIdx-- )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx – 1]
RefPicListX[ refIdxLX++ ] = inter-view prediction reference with view_id equal to targetViewID
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_lX_active_minus1 + 1; cIdx++ )                    (H-8)
    if( viewID(RefPicListX[ cIdx ]) != targetViewID | | PictureOrderCnt(RefPicListX[ cIdx ]) != currPOC )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]
```

In the above, the function viewID(refpic) returns the view_id of the reference picture refpic, the variable currViewID is equal to the view_id of the current slice, and the variable currPOC is equal to PicOrderCnt() of the current slice.

H.8.3 MVC decoded reference picture marking process

The specification in subclause 8.2.5 shall apply independently for each view, with "picture" being replaced by "view component", "frame" being replaced by "frame view component", and "field" being replaced by "field view component".

H.8.4 MVC inter prediction and inter-view prediction process

For both inter-prediction and inter-view prediction, the specification in subclause 8.4 shall apply.

H.8.5 Specification of bitstream subsets

Subclauses H.8.5.1 and H.8.5.2 specify the processes for deriving required anchor and non-anchor view components, respectively, that are used in the sub-bitstream extraction process. Subclause H.8.5.3 specifies the sub-bitstream extraction process. Subclause H.8.5.4 specifies the base view bitstream subset. Subclause H.8.5.5 gives an informative example for creation of a base view in case the orignal base view in the input bitstream to the bitstream extraction process is not included in the output bitstream subset.

H.8.5.1 Derivation process for required anchor view components

This process is recursively invoked to derive the set of required anchor view components for a specified view. The view_id's of all views for which the anchor view components are required for the specified view are marked as "required for anchor" and their corresponding VOIdx values are included in VOIdxList.

Input to this process is a variable viewId, representing a view with view_id equal to viewId, with its corresponding view order index denoted by vOIdx.

Outputs of this process are the view_id equal to viewId being marked as "required for anchor", a possibly updated VOIdxList, and additional invocations of the derivation process based on the inter-view dependency for anchor view components in the view with view_id equal to viewId as specified in the sequence parameter set MVC extension.

The following process is applied:

- Mark the view_id equal to viewId as "required for anchor" and add vOIdx to VOIdxList if the same value is not already included in VOIdxList.
- If both num_anchor_refs_l0[vOIdx] and num_anchor_refs_l1[vOIdx] are equal to 0, terminate this process.
- Otherwise (num_anchor_refs_l0[vOIdx] or num_anchor_refs_l1[vOIdx] is not equal to 0), the following is applied:
    - When num_anchor_refs_l0[vOIdx] is not equal to 0, invoke the process specified in subclause H.8.5.1 for each viewId equal to anchor_ref_l0[vOIdx][i] for all i in the range of 0 to num_anchor_refs_l0[vOIdx] – 1, inclusive, in ascending order of i.
    - When num_anchor_refs_l1[vOIdx] is not equal to 0, invoke the process specified in subclause H.8.5.1 for each viewId equal to anchor_ref_l1[vOIdx][i] for all i in the range of 0 to num_anchor_refs_l1[vOIdx] – 1, inclusive, in ascending order of i.

H.8.5.2 Derivation process for required non-anchor view components

This process is recursively invoked to derive the set of required non-anchor view components for a specified view. The view_id's of all views for which the non-anchor view components are required for the specified view are marked as "required for non-anchor".

Input to this process is a variable viewId, representing a view with view_id equal to viewId, with its corresponding view order index denoted by vOIdx.

Outputs of this process are the view_id equal to viewId being marked as "required for non-anchor" and additional invocations of the derivation process based on the inter-view dependency for non-anchor view components in the view with viwe_id equal to viewId as specified in the sequence parameter set MVC extension.

The following process is applied:

– Mark the view_id equal to viewId as "required for non-anchor".

– If both num_non_anchor_refs_l0[vOIdx] and num_non_anchor_refs_l1[vOIdx] are equal to 0, terminate this process.

– Otherwise (num_non_anchor_refs_l0[vOIdx] or num_non_anchor_l1[vOIdx] is not equal to 0), the following is applied:

  – When num_non_anchor_refs_l0[vOIdx] is not equal to 0, invoke the process specified in subclause H.8.5.2 for each viewId equal to non_anchor_ref_l0[vOIdx][i] for all i in the range of 0 to num_non_anchor_l0[vOIdx] – 1, inclusive, in ascending order of i.

  – When num_non_anchor_refs_l1[vOIdx] is not equal to 0, invoke the process specified in subclause H.8.5.2 for each viewId equal to non_anchor_ref_l1[vOIdx][i] for all i in the range of 0 to num_non_anchor_l1[vOIdx] – 1, inclusive, in ascending order of i.

H.8.5.3 Sub-bitstream extraction process

It is requirement of bitstream conformance that any sub-bitstream that is the output of the process specified in this subclause with pIdTarget equal to any value in the range of 0 to 63, inclusive, tIdTarget equal to any value in the range of 0 to 7, inclusive, viewIdTargetList consisting of any one or more viewIdTarget's identifying the views in the bitstream, shall be conforming to this Recommendation | International Standard.

NOTE 1 – A conforming bitstream contains one or more coded slice NAL units with priority_id equal to 0 and temporal_id equal to 0.

NOTE 2 – It is possible that not all operation points of sub-bitstreams resulting from the sub-bitstream extraction process have an applicable level_idc or level_idc[i]. In this case, each coded video sequence in a sub-bitstream must still conform to one or more of the profiles specified in Annex A and Annex H, but may not satisfy the level constraints specified in subclause A.3 and H.10.2, respectively.

Inputs to this process are

– a variable pIdTarget (when present),

– a variable tIdTarget (when present),

– a list viewIdTargetList consisting of one or more viewIdTarget's (when present).

Outputs of this process are a sub-bitstream and a list of VOIdx values VOIdxList.

When pIdTarget is not present as input to this subclause, pIdTarget is inferred to be equal to 63.

When tIdTarget is not present as input to this subclause, tIdTarget is inferred to be equal to 7.

When viewIdTargetList is not present as input to this subclause, there shall be one viewIdTarget inferred in viewIdTargetList and the value of viewIdTarget is inferred to be equal to view_id of the base view.

The sub-bitstream is derived by applying the following operations in sequential order.

1. Let VOIdxList be empty and minVOIdx be the VOIdx value of the base view.
2. For each viewIdTarget included in viewIdTargetList, invoke the process specified in subclause H.8.5.1 with the viewIdTarget as input.
3. For each viewIdTarget included in viewIdTargetList, invoke the process specified in subclause H.8.5.2 with the viewIdTarget as input.
4. Mark all VCL NAL units and filler data NAL units for which any of the following conditions is true as "to be removed from the bitstream".
   – priority_id is greater than pIdTarget
   – temporal_id is greater than tIdTarget

– anchor_pic_flag is equal to 1 and view_id is not marked as "required for anchor"

– anchor_pic_flag is equal to 0 and view_id is not marked as "required for non-anchor"

– nal_ref_idc is equal to 0 and inter_view_flag is equal to 0 and view_id is not equal to any value in the list OutputVOIdxList

5. Remove all access units for which all VCL NAL units are marked as "to be removed from the bitstream".

6. Remove all VCL NAL units and filler data NAL units that are marked as "to be removed from the bitstream".

7. When VOIdxList contains only one value of VOIdx that is equal to minVOIdx, remove the following NAL units.

   NOTE 3 – When VOIdxList contains only one value of VOIdx equal to minVOIdx, the sub-bitstream contains only the base view or only a temporal subset of the base view.

   – all NAL units with nal_unit_type equal to 14 or 15

   – all NAL units with nal_unit_type equal to 6 in which the first SEI message has payloadType in the range of 36 to 44, inclusive

8. Let maxTId be the maximum temporal_id of all the remaining VCL NAL units. Remove all NAL units with nal_unit_type equal to 6 that only contain SEI messages that are part of an MVC scalable nesting SEI message with any of the following properties.

   – operation_point_flag is equal to 0 and all_view_components_in_au_flag is equal to 0 and none of sei_view_id[i] for all i in the range of 0 to num_view_components_minus1, inclusive, corresponds to a VOIdx value included in VOIdxList

   – operation_point_flag is equal to 1 and either sei_op_temporal_id is greater than maxTId or the list of sei_op_view_id[i] for all i in the range of 0 to num_view_components_op_minus1, inclusive, is not a subset of viewIdTargetList (i.e. it is not ture that sei_op_view_id[i] for any i in the range of 0 to num_view_components_op_minus1, inclusive, is equal to a value in viewIdTargetList)

9. Remove each view scalability information SEI message and each operation point not present SEI message, when present.

10. When VOIdxList does not contain a value of VOIdx equal to minVOIdx, the view with VOIdx equal to the mininum VOIdx value included in VOIdxList shall be converted to the base view of the extracted sub-bitstream. An informative procedure that outlines key processing steps to create a base view is described in H.8.5.5.

    NOTE 4 – When VOIdxList does not contain a value of VOIdx equal to minVOIdx, the resulting sub-bitstream according to the operation steps 1-9 above does not contain a base view that conforms to one or more profiles specified in Annex A. In this case, by this operation step, the remaining view with the new minimum VOIdx value is converted to be the new base view that conforms to one or more profiles specified in Annex A.

H.8.5.4 Specification of the base view bitstream

A bitstream that conforms to one or more profiles as specified in Annex H shall contain a base view bitstream that conforms to one or more of the profiles specified in Annex A. This base view bitstream is derived by invoking the sub-bitstream extraction process as specified in subclause H.8.5.3 with no input and the base view bitstream being the output.

H.8.5.5 Creation of a base view during sub-bitstream extraction (informative)

According to the sub-bitstream extraction process specified in H.8.5.3, the resulting sub-bitstream shall contain a base view. When the resulting bistream does not contain a base view, the following procedure may be used to create a base view during sub-bitstream extraction.

When VOIdxList does not contain a value of VOIdx equal to minVOIdx, let newBaseViewId be equal to the view_id for which the VOIdx value is equal to the mininum VOIdx value included in VOIdxList, and apply the following operations in sequential order.

– Remove all NAL units with nal_unit_type equal to 7.

– For all subset sequence parameter set NAL units (with nal_unit_type equal to 15) that are referred to by at least one remaining VCL NAL unit with view_id equal to newBaseViewId, apply the following operations in sequential order.

- Set nal_unit_type to 7.
- Set profile_idc to 100.
- Set level_idc to level_idc[i], with i equal to the value that for one value of j in the range of 0 to num_applicable_ops_minus1[i], inclusive, applicable_op_temporal_id[i][j] is equal to maxTId, applicable_op_num_target_views_minus1[i][j] is equal to 0, and applicable_op_target_view_id[i][j][k] for k equal to 0 is equal to newBaseViewId.
- Remove all the syntax elements after the syntax structure seq_parameter_set_data( ) and before the syntax structure rbsp_trailing_bits( ), and change RBSP trailing bits appropriately.

– Remove all SEI NAL units (with nal_unit_type equal to 6) for which the first contained SEI message has payloadType in the range of 0 to 23, inclusive.

– For each SEI NAL unit (with nal_unit_type equal to 6) containing an MVC scalable nesting SEI message, the following operations are applied in sequential order.

- When none of the following properties is true for the MVC scalable nesting SEI message, the SEI NAL unit is removed.
- operation_point_flag is equal to 0 and all_view_components_in_au_flag is equal to 1
- operation_point_flag is equal to 0, all_view_components_in_au_flag is equal to 0, and at least one of the values of sei_view_id[i] for all i in the range of 0 to num_view_components_minus1, inclusive, is equal to one of the viewIdTarget's in viewIdTargetList
- operation_point_flag is equal to 1, sei_op_temporal_id is equal to or less than maxtIdT, and the list of sei_op_view_id[i] for all i in the range of 0 to num_view_components_op_minus1, inclusive, is a subset of viewIdTargetList (i.e. it is true that sei_op_view_id[i] for any i in the range of 0 to num_view_components_op_minus1, inclusive, is equal to a value in viewIdTargetList)
- When the SEI NAL unit is not removed, the following applies:
- If VOIdxList contains only one VOIdx value, the SEI NAL unit is replaced by an SEI NAL unit containing only the original nested SEI message not as part of an MVC scalable nesting SEI message.
- Otherwise (VOIdxList contains more than one VOIdx value), when any of the the following properties is true for the MVC scalable nesting SEI message, a new SEI NAL unit containing only the nested SEI message not as part of an MVC scalable nesting SEI message is generated and inserted immediately before the original SEI NAL unit in decoding order.
    - operation_point_flag is equal to 0 and all_view_components_in_au_flag is equal to 1
    - operation_point_flag is equal to 0, all_view_components_in_au_flag is equal to 0, and for the values of sei_view_id[i] for all i in the range of 0 to num_view_components_minus1, inclusive, one is equal to newBaseViewId, and at least another one is equal to one of the viewIdTarget's in viewIdTargetList

– When VOIdxList contains only one value of VOIdx, remove the following NAL units.

- All NAL units with nal_unit_type equal to 15
- All NAL units with nal_unit_type equal to 6 in which the first SEI message has payloadType in the range of 36 to 44, inclusive

– For each NAL unit nalUnit with nal_unit_type equal to 20 and view_id equal to newBaseViewId, the following operations are applied in sequential order.

- If idr_flag is equal to 1, set nal_unit_type equal to 5. Otherwise (idr_flag is equal to 0), set nal_unit_type equal to 1.
- When VOIdxList contains more than one VOIdx value, generate a prefix NAL unit with the same NAL unit header (including NAL unit header extension) as the NAL unit nalUnit, except that nal_unit_type is set to 14 and priority_id may be changed, and insert the prefix NAL unit immediately before the NAL unit nalUnit in decoding order. After the last application of this operation, at least one of all the inserted prefix NAL units by the applications of this operation shall have priority_id equal to 0.

– Remove the NAL unit header extension of nalUnit.

H.9 Parsing process

The specifications in clause 9 apply.

H.10 Profiles and levels

H.10.1 Multiview High profile

Bitstreams conforming to the Multiview High profile shall obey the following constraints:

– The base view bitstream as specified in subclause H.8.5.4 shall obey all the constraints of the High profile specified in subclause A.2.4.

– Only I, P, and B slice types may be present.

– NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive.

– MVC sequence parameter sets shall have frame_mbs_only_flag equal to 1.

– Arbitrary slice order is not allowed.

– Picture parameter sets shall have num_slice_groups_minus1 equal to 0 only.

– Picture parameter sets shall have redundant_pic_cnt_present_flag equal to 0 only.

– MVC sequence parameter sets shall have chroma_format_idc in the range of 0 to 1 inclusive.

– MVC sequence parameter sets shall have bit_depth_luma_minus8 equal to 0 only.

– MVC sequence parameter sets shall have bit_depth_chroma_minus8 equal to 0 only.

– MVC sequence parameter sets shall have qpprime_y_zero_transform_bypass_flag equal to 0 only.

– The level constraints specified for the Multiview High profile in subclause H.10.2 shall be fulfilled.

Conformance of a bitstream to the Multiview High profile is specified by profile_idc being equal to 118.

Decoders conforming to the Multiview High profile at a specific level shall be capable of decoding all bitstreams in which both of the following conditions are true:

– All active MVC sequence parameter sets have any of the following.
  – profile_idc equal to 118.
  – profile_idc equal to 100 or 77 and constraint_set4_flag is equal to 1.
  – profile_idc equal to 88 and constraint_set1_flag equal to 1 and constraint_set4_flag is equal to 1.
  – profile_idc equal to 66 and constraint_set1_flag equal to 1.

– All active MVC sequence parameter sets have any of the following.
  – level_idc or (level_idc and constraint_set3_flag) represent a level less than or equal to the specific level.
  – level_idc[i] or (level_idc[i] and constraint_set3_flag) represent a level less than or equal to the specific level.

NOTE – When profile_idc is equal to 100, 77 or 88 and constraint_set4_flag is equal to 1, the bitstream conforms to the High profile and additionally conforms to the constraints specified for the Multiview High profile specified in this subclause. For example, such a bitstream must have frame_mbs_only_flag equal to 1.

H.10.2 Levels

The following is specified for expressing the constraints in this Annex.

– Let access unit n be the n-th access unit in decoding order with the first access unit being access unit 0.

– Let picture n be the primary coded picture or the corresponding decoded picture of access unit n.

Let the variable fR be derived as follows.

– If picture n is a frame, fR is set equal to 1 ÷ 172.

– Otherwise (picture n is a field), fR is set equal to 1 ÷ (172 * 2).

The value of mvcScaleFactor is set equal to 2.

The value of NumViews is set equal to applicable_op_num_views_minus1[i][j], which indicates the number of views for the j-th operation point for level_idc[i] as signaled in the subset sequence parameter set.

Bitstreams conforming to the Multiview High profile at a specified level shall obey the following constraints:

a) The nominal removal time of access unit n (with n > 0) from the CPB as specified in subclause C.1.2, satisfies the constraint that $t_{r,n}(n) - t_r(n - 1)$ is greater than or equal to Max(NumViews * PicSizeInMbs ÷ (mvcScaleFactor * MaxMBPS), fR), where MaxMBPS is the value specified in Table A-1 that applies to picture n – 1, and PicSizeInMbs is the number of macroblocks in a single view component of picture n – 1.

b) The difference between consecutive output times of pictures from the DPB as specified in subclause C.2.2, satisfies the constraint that $\Delta t_{o,dpb}(n)$ >= Max(NumViews * PicSizeInMbs ÷ (mvcScaleFactor * MaxMBPS), fR), where MaxMBPS is the value specified in Table A-1 for picture n, and PicSizeInMbs is the number of macroblocks of a single view component of picture n, provided that picture n is a picture that is output and is not the last picture of the bitstream that is output.

c) PicWidthInMbs * FrameHeightInMbs <= MaxFS, where MaxFS is specified in Table A-1

d) PicWidthInMbs <= Sqrt(MaxFS * 8)

e) FrameHeightInMbs <= Sqrt(MaxFS * 8)

f) max_dec_frame_buffering <= MaxDpbFrames, where MaxDpbFrames is equal to Min(mvcScaleFactor * MaxDpbMbs / (PicWidthInMbs * FrameHeightInMbs), Max(1, Ceil(log2(NumViews))) * 16) and MaxDpbMbp is specified in Table A-1.

g) Vertical motion vector component range does not exceed MaxVmvR in units of luma frame samples, where MaxVmvR is specified in Table A-1.

h) Horizontal motion vector range does not exceed the range of −2048 to 2047.75, inclusive, in units of luma samples.

i) Let setOf2Mb be the set of unsorted pairs of macroblocks that contains the unsorted pairs of macroblocks (mbA, mbB) of a coded video sequence for which any of the following conditions is true.

   – mbA and mbB are macroblocks that belong to the same slice and are consecutive in decoding order

   – separate_colour_plane_flag is equal to 0, mbA is the last macroblock (in decoding order) of a slice, and mbB is the first macroblock (in decoding order) of the next slice in decoding order

   – separate_colour_plane_flag is equal to 1, mbA is the last macroblock (in decoding order) of a slice with a particular value of colour_plane_id, and mbB is the first macroblock (in decoding order) of the next slice with the same value of colour_plane_id in decoding order

   NOTE 1 – In the two above conditions, the macroblocks mbA and mbB can belong to different pictures.

   For each unsorted pair of macroblocks (mbA, mbB) of the set setOf2Mb, the total number of motion vectors (given by the sum of the number of motion vectors for macroblock mbA and the number of motion vectors for macroblock mbB) does not exceed MaxMvsPer2Mb, where MaxMvsPer2Mb is specified in Table A-1. The number of motion vectors for each macroblock is the value of the variable MvCnt after the completion of the intra or inter prediction process for the macroblock.

   NOTE 2 – When separate_colour_plane_flag is equal to 0, the constraint specifies that the total number of motion vectors for two consecutive macroblocks in decoding order must not exceed MaxMvsPer2Mb. When separate_colour_plane_flag is equal to 1, the constraint specifies that the total number of motion vectors for two consecutive macroblocks with the same value of colour_plane_id in decoding order must not exceed MaxMvsPer2Mb. For macroblocks that are consecutive in decoding order but are associated with a different value of colour_plane_id, no constraint for the total number of motion vectors is specified.

j) Number of bits of macroblock_layer( ) data for any macroblock is not greater than 128 + RawMbBits. Depending on entropy_coding_mode_flag, the bits of macroblock_layer( ) data are counted as follows.
   – If entropy_coding_mode_flag is equal to 0, the number of bits of macroblock_layer( ) data is given by the number of bits in the macroblock_layer( ) syntax structure for a macroblock.
   – Otherwise (entropy_coding_mode_flag is equal to 1), the number of bits of macroblock_layer( ) data for a macroblock is given by the number of times read_bits(1) is called in subclauses 9.3.3.2.2 and 9.3.3.2.3 when parsing the macroblock_layer( ) associated with the macroblock.

k) In bitstreams conforming to the Multiview High profile, the difference between consecutive removal time of access units n and n - 1 with n > 0 shall satisfy the constraint that the number of slices in picture n is less than or equal to mvcScaleFactor * MaxMBPS * (t_r(n) - t_r(n - 1)) ÷ SliceRate, where SliceRate is the value specified in Table A-4 that applies to picture n.

l) In bitstreams conforming to the Multiview High profile, MVC sequence parameter sets shall have direct_8x8_inference_flag equal to 1 for the levels specified in Table A-4.

m) In bitstreams conforming to the Multiview High profile, MVC sequence parameter sets shall have frame_mbs_only_flag equal to 1 for all levels.

n) In bitstreams conforming to the Multiview High profile, the value of sub_mb_type[mbPartIdx] with mbPartIdx = 0..3 in B macroblocks with mb_type equal to B_8x8 shall not be equal to B_Bi_8x4, B_Bi_4x8, or B_Bi_4x4 for the levels in which MinLumaBiPredSize is shown as 8x8 in Table A-4.

o) In bitstreams conforming to the Multiview High profile, for the VCL HRD parameters, BitRate[SchedSelIdx] <= cpbBrVclFactor * MaxBR and CpbSize[SchedSelIdx] <= cpbBrVclFactor * MaxCPB for at least one value of SchedSelIdx, where cpbBrVclFactor is equal to 1250, and BitRate[SchedSelIdx] and CpbSize[SchedSelIdx] are given as follows.

   – If vcl_hrd_parameters_present_flag equal to 1, BitRate[SchedSelIdx] and CpbSize[SchedSelIdx] are given by Equations E-37 and E-38, respectively, using the syntax elements of the hrd_parameters( ) syntax structure that immediately follows vcl_hrd_parameters_present_flag.

   – Otherwise, BitRate[SchedSelIdx] and CpbSize[SchedSelIdx] are inferred as specified in subclause E.2.2 for VCL HRD parameters.

   MaxBR and MaxCPB are specified in Table A-1 in units of cpbBrVclFactor bits/s and cpbBrVclFactor bits, respectively. The bitstream shall satisfy these conditions for at least one value of SchedSelIdx in the range 0 to cpb_cnt_minus1, inclusive.

p) In bitstreams conforming to the Multiview High profile, for the NAL HRD parameters, BitRate[SchedSelIdx] <= cpbBrNalFactor * MaxBR and CpbSize[SchedSelIdx] <= cpbBrNalFactor * MaxCPB for at least one value of SchedSelIdx, where cpbBrNalFactor is equal to 1500, and BitRate[SchedSelIdx] and CpbSize[SchedSelIdx] are given as follows.

   – If nal_hrd_parameters_present_flag equal to 1, BitRate[SchedSelIdx] and CpbSize[SchedSelIdx] are given by Equations E-37 and E-38, respectively, using the syntax elements of the hrd_parameters( ) syntax structure that immediately follows nal_hrd_parameters_present_flag.

   – Otherwise, BitRate[SchedSelIdx] and CpbSize[SchedSelIdx] are inferred as specified in subclause E.2.2 for NAL HRD parameters.

   MaxBR and MaxCPB are specified in Table A-1 in units of cpbBrNalFactor bits/s and cpbBrNalFactor bits, respectively. The bitstream shall satisfy these conditions for at least one value of SchedSelIdx in the range 0 to cpb_cnt_minus1, inclusive.

q) In bitstreams conforming to the Multiview High profile, the sum of the NumBytesInNALunit variables for access unit n with n > 0 is less than or equal to 384 * mvcScaleFactor * MaxMBPS * (t_r(n) - t_r(n – 1)) ÷ MinCR, where MaxMBPS and MinCR are the values specified in Table A-1 that apply to picture n.

r) In bitstreams conforming to the Multiview High profile, when PicSizeInMbs is greater than 1620, the number of macroblocks in any coded slice shall not exceed MaxFS / 4, where MaxFS is specified in Table A-1.

Table A-1 specifies the limits for each level. Entries marked "-" in Table A-1 denote the absence of a corresponding limit. The use of the MinCR parameter column of Table A-1 for these profiles is specified in subclause A.3.3.

For coded video sequences conforming to the Multiview High profile, the level_idc value is specified as follows:

– If level_idc is not equal to 0, level_idc indicates the level that applies to the coded video sequence operating with all the views being target output views.

  NOTE 1 – A level_idc value that is not equal to zero may indicate a higher level than necessary to decode the coded video sequence operating with all the views being target output views. This may occur when a subset of views or temporal subsets are removed from a coded video sequence according to the sub-bitstream extraction process specified in H.8.5.3, and the level_idc value is not updated accordingly.

– Otherwise (level_idc is equal to 0), the level that applies to the coded video sequence operating with all the views being target output views is unspecified.

NOTE 2 – When profile_idc is equal to 118 and level_idc is equal to 0, there may exist a level indicated by level_idc[i] that is applicable to the coded video sequence operating with all the views being target output views. This may occur when a subset of views or temporal subsets are removed from a coded video sequence according to the sub-bitstream extraction process specified in H.8.5.3, and a particular value of level_idc[i] corresponds to the resulting coded video sequence.

A level to which the bitstream conforms shall be indicated by the syntax element level_idc or level_idc[i] as follows.

– If level_idc or level_idc[i] is equal to 9, the indicated level is level 1b.

– Otherwise (level_idc or level_idc[i] is not equal to 9), the indicated level number is equal to level_idc or level_idc[i] divided by 10.

Table A-4 specifies limits for each level that are specific to bitstreams conforming to the Multiview High profile. Entries marked "-" in Table A-4 denote the absence of a corresponding limit.

H.11 MVC hypothetical reference decoder

The specifications in Annex C apply with substituting MVC sequence parameter set for sequence parameter set.

H.12 Byte stream format

The specifications in Annex B apply.

H.13 MVC SEI messages

The specifications in Annex D together with the extensions and modifications specified in this subclause apply.

H.13.1 SEI message syntax

H.13.1.1 Parallel decoding information SEI message syntax

parallel_decoding_info(payloadSize) {	C	Descriptor
seq_parameter_set_id	5	ue(v)
for(i = 1; i <= num_views_minus1; i++) {		
if(anchor_pic_flag) {		
for(j = 0; j <= num_anchor_refs_l0[i]; j++)		
pdi_init_delay_anchor_minus2_l0[i **][** j **]**	5	ue(v)
for(j = 0; j <= num_anchor_refs_l1[i]; j++)		
pdi_init_delay_anchor_minus2_l1[i **][** j **]**	5	ue(v)
}		
else {		
for(j = 0; j <= num_non_anchor_refs_l0[i]; j++)		
pdi_init_delay_non_anchor_minus2_l0[i **][** j **]**	5	ue(v)
for(j = 0; j <= num_non_anchor_refs_l1[i]; j++)		
pdi_init_delay_non_anchor_minus2_l1[i **][** j **]**	5	ue(v)
}		
}		
}		

H.13.1.2 MVC scalable nesting SEI message syntax

mvc_scalable_nesting(payloadSize) {	C	Descriptor
operation_point_flag	5	u(1)
if (!operation_point_flag) {		
all_view_components_in_au_flag	5	u(1)

if(!all_view_components_in_au_flag) {		
num_view_components_minus1	5	ue(v)
for(i = 0; i <= num_view_components_minus1; i++)		
sei_view_id[i **]**	5	u(10)
} else {		
num_view_components_op_minus1	5	ue(v)
for(i = 0; i <= num_view_components_op_minus1; i++)		
sei_op_view_id[i **]**	5	u(10)
sei_op_temporal_id	5	u(3)
}		
while(!byte_aligned( ))		
sei_nesting_zero_bit /* equal to 0 */	5	f(1)
sei_message( )	5	
}		

H.13.1.3 View scalability information SEI message syntax

view_scalability_info(payloadSize) {	C	Descriptor
num_operation_points_minus1	5	ue(v)
for(i = 0; i <= num_operation_points_minus1; i++) {		
operation_point_id[i **]**	5	ue(v)
priority_id[i **]**	5	u(5)
temporal_id[i **]**	5	u(3)
num_target_output_views_minus1[i **]**	5	ue(v)
for(j = 0; j <= num_target_output_views_minus1[i]; j++)		
view_id[i **][** j **]**	5	ue(v)
profile_level_info_present_flag[i **]**	5	u(1)
bitrate_info_present_flag[i **]**	5	u(1)
frm_rate_info_present_flag[i **]**	5	u(1)
if(!num_target_output_views_minus1[i])		
view_dependency_info_present_flag[i **]**	5	u(1)
parameter_sets_info_present_flag[i **]**	5	u(1)
bitstream_restriction_info_present_flag[i]	5	u(1)
if (profile_level_info_present_flag[i])		
op_profile_level_idc[i **]**	5	u(24)
if(bitrate_info_present_flag[i]) {		
avg_bitrate[i **]**	5	u(16)
max_bitrate[i **]**	5	u(16)
max_bitrate_calc_window[i **]**	5	u(16)
}		
if(frm_rate_info_present_flag[i]) {		
constant_frm_rate_idc[i **]**	5	u(2)
avg_frm_rate[i **]**	5	u(16)
}		
if(view_dependency_info_present_flag[i]) {		
num_directly_dependent_views[i **]**	5	ue(v)
for(j = 0; j < num_directly_dependent_views[i]; j++) {		

directly_dependent_view_id[i **][** j **]**	5	ue(v)
} else		
view_dependency_info_src_op_id[i **]**	5	ue(v)
if(parameter_sets_info_present_flag[i]) {		
num_seq_parameter_set_minus1[i **]**	5	ue(v)
for(j = 0; j <= num_seq_parameter_set_minus1[i]; j++)		
seq_parameter_set_id_delta[i **][** j **]**	5	ue(v)
num_subset_seq_parameter_set_minus1[i **]**	5	ue(v)
for(j = 0; j <= num_subset_seq_parameter_set_minus1[i]; j++)		
subset_seq_parameter_set_id_delta[i **][** j **]**	5	ue(v)
num_pic_parameter_set_minus1[i **]**	5	ue(v)
for(j = 0; j <= num_init_pic_parameter_set_minus1[i]; j++)		
pic_parameter_set_id_delta[i **][** j **]**	5	ue(v)
} else		
parameter_sets_info_src_op_id[i **]**	5	ue(v)
if(bitstream_restriction_info_present_flag[i]) {		
motion_vectors_over_pic_boundaries_flag[i **]**	5	u(1)
max_bytes_per_pic_denom[i **]**	5	ue(v)
max_bits_per_mb_denom[i **]**	5	ue(v)
log2_max_mv_length_horizontal[i **]**	5	ue(v)
log2_max_mv_length_vertical[i **]**	5	ue(v)
num_reorder_frames[i **]**	5	ue(v)
max_dec_frame_buffering[i **]**	5	ue(v)
}		
}		
}		

H.13.1.4 Multiview scene information SEI message syntax

multiview_scene_info(payloadSize) {	**C**	**Descriptor**
max_disparity	5	ue(v)
}		

H.13.1.5 Multiview acquisition information SEI message syntax

multiview_acquisition_info(payloadSize) {	**C**	**Descriptor**
num_views_minus1		ue(v)
intrinsic_param_flag	5	u(1)
extrinsic_param_flag	5	u(1)
if (instrinsic_param_flag) {		
i**ntrinsic_params_equal**	5	u(1)
prec_focal_length	5	ue(v)
prec_principal_point	5	ue(v)
prec_skew_factor	5	ue(v)
if(intrinsic_params_equal)		
num_of_param_sets = 1		

else		
num_of_param_sets = num_views_minus1 + 1		
for(i = 0; i < num_of_param_sets; i++) {		
sign_focal_length_x[i **]**	5	u(1)
exponent_focal_length_x[i **]**	5	u(6)
mantissa_focal_length_x[i **]**	5	u(v)
sign_focal_length_y[i **]**	5	u(1)
exponent_focal_length_y[i **]**	5	u(6)
mantissa_focal_length_y[i **]**	5	u(v)
sign_principal_point _x[i **]**	5	u(1)
exponent_principal_point_x[i **]**	5	u(6)
mantissa_principal_point_x[i **]**	5	u(v)
sign_principal_point _y[i **]**	5	u(1)
exponent_principal_point_y[i **]**	5	u(6)
mantissa_principal_point_y[i **]**	5	u(v)
sign_skew_factor[i **]**	5	u(1)
exponent_skew_factor[i **]**	5	u(6)
mantissa_skew_factor[i **]**	5	u(v)
}		
}		
if(extrinsic_param_flag) {		
prec_rotation_param	5	ue(v)
prec_translation_param	5	ue(v)
for(i = 0; i <= num_views_minus1; i++) {		
for (j = 1; j <= 3; j++) { /* row */		
for (k = 1; k <= 3; k++) { /* column */		
sign_r[i **][** j **][** k **]**	5	u(1)
exponent_r[i **][** j **][** k **]**	5	u(6)
mantissa_r[i **][** j **][** k **]**	5	u(v)
}		
sign_t[i **][** j **]**	5	u(1)
exponent_t[i **][** j **]**	5	u(6)
mantissa_t[i **][** j **]**	5	u(v)
}		
}		
}		
}		

H.13.1.6 Non-required view component SEI message syntax

non_required_view_component(payloadSize) {	C	Descriptor
num_info_entries_minus1	5	ue(v)
for(i = 0; i <= num_info_entries_minus1; i++) {		
view_order_index[i **]**	5	ue(v)
num_non_required_view_components_minus1[i **]**	5	ue(v)
for(j = 0; j <= num_non_required_view_components_minus1[i]; j++)		
index_delta_minus1[i **][** j **]**	5	ue(v)
}		
}		

H.13.1.7 View dependency change SEI message syntax

view_dependency_change(payloadSize) {	C	Descriptor
seq_parameter_set_id	5	ue(v)
anchor_update_flag	5	u(1)
non_anchor_update_flag	5	u(1)
if(anchor_update_flag)		
for(i = 1; i <= num_views_minus1; i++) {		
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0_flag[i **][** j **]**	5	u(1)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1_flag[i **][** j **]**	5	u(1)
}		
if(non_anchor_update_flag)		
for(i = 1; i <= num_views_minus1; i++) {		
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0_flag[i **][** j **]**	5	u(1)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1_flag[i **][** j **]**	5	u(1)
}		
}		

H.13.1.8 Operation point not present SEI message syntax

operation_point_not_present(payloadSize) {	C	Descriptor
num_operation_points	5	ue(v)
for(k = 0; k < num_operation_points; k++)		
operation_point_not_present_id[k **]**	5	ue(v)
}		

H.13.1.9 Base view temporal HRD SEI message syntax

base_view_temporal_hrd (payloadSize) {	C	Descriptor
num_of_temporal_layers_in_base_view_minus1	5	ue(v)
for(i = 0; i < num_of_temporal_layers_in_base_view_minus1; i++) {		
temporal_id[i **]**	5	u(3)
timing_info_present_flag[i **]**	5	u(1)
if(timing_info_present_flag[i]) {		
num_units_in_tick[i **]**	5	u(32)
time_scale[i **]**	5	u(32)
fixed_frame_rate_flag[i **]**	5	u(1)
}		
nal_hrd_parameters_present_flag[i **]**	5	u(1)
if(nal_hrd_parameters_present_flag[i])		
hrd_parameters( )		
vcl_hrd_parameters_present_flag[i **]**	5	u(1)
if(vcl_hrd_parameters_present_flag[i])		
hrd_parameters( )		
if(nal_hrd_parameters_present_flag[i] || vcl_hrd_parameters_present_flag[i])		
low_delay_hrd_flag[i **]**	5	u(1)
pic_struct_present_flag[i **]**	5	u(1)
}		
}		

H.13.2 SEI message semantics

Depending on payloadType, the corresponding SEI message semantics are extended as follows.

– If payloadType is equal to 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 or 23, the following applies:

  – If the SEI message is not included in an MVC scalable nesting SEI message, it applies to the view component of the current access unit with VOIdx equal to VOIdxMin.

  – Otherwise (the SEI message is included in an MVC scalable nesting SEI message), it applies to all view components of the current access unit when all_view_components_in_au_flag is equal to 1, or it applies to all view components of the current access unit with view_id equal to sei_view_id[i] for any i in the range of 0 to num_view_components_minus1, inclusive, when all_view_components_in_au_flag is equal to 0. When payloadType is equal to 10 for the SEI message that is included in an MVC scalable nesting SEI message, the semantics for sub_seq_layer_num of the sub-sequence information SEI message is modified as follows.

    sub_seq_layer_num specifies the sub-sequence layer number of the current picture. When the current picture resides in a sub-sequence whose first picture in decoding order is an IDR picture, the value of sub_seq_layer_num shall be equal to 0. For a non-paired reference field, the value of sub_seq_layer_num shall be equal to 0. sub_seq_layer_num shall be in the range of 0 to 255, inclusive.

– Otherwise, if payloadType is equal to 0 or 1, the following applies:

  – If the SEI message is not included in an MVC scalable nesting SEI message, the following applies. When the SEI message and all other SEI messages with payloadType equal to 0 or 1 not included in an MVC scalable nesting SEI message are used as the buffering period and picture timing SEI messages for checking the bitstream conformance according to Annex C and the decoding process specified in clauses 2-9 of this Recommendation | International Standard is used, the bitstream shall be conforming to this Recommendation | International Standard.

– Otherwise (the SEI message is included in an MVC scalable nesting SEI message), the following applies. When the SEI message and all other SEI messages with payloadType equal to 0 or 1 included in an MVC scalable nesting SEI message with identical values of sei_op_temporal_id and sei_op_view_id[i] for all i in the range of 0 to num_view_components_op_minus1, inclusive, are used as the buffering period and picture timing SEI messages for checking the bitstream conformance according to Annex C, the bitstream that would be obtained by invoking the bitstream extraction process as specified in subclause H.8.3 with tIdTarget equal to sei_op_temporal_id and viewIdTargetList equal to sei_op_view_id[i] for all i in the range of 0 to num_view_components_op_minus1, inclusive, shall be conforming to this Recommendation | International Standard.

  The values of seq_parameter_set_id's in all buffering period SEI messages included in MVC scalable nesting SEI messages and associated with operation points for which the greatest VOIdx values in the associated bitstream sbusets are identical shall be identical.

– Otherwise (all remaining payloadType values), the corresponding SEI message semantics are not extended.

For the semantics of SEI messages with payloadType in the range of 0 to 23, inclusive, which are specified in subclause D.2, MVC sequence parameter set is substituted for sequence paramete set; the parameters of MVC sequence parameter set RBSP and picture parameter set RBSP that are in effect are specified in subclauses H.7.4.2.1 and H.7.4.2.2, respectively.

Coded video sequences conforming to one or more of the profiles specified in Annex H shall not include SEI NAL units that contain SEI messages with payloadType in the range of 24 to 35, inclusive.

When an SEI NAL unit contains an SEI message with payloadType in the range of 36 to 44, inclusive, it shall not contain any SEI messages with payloadType less than 36 and the first SEI message in the SEI NAL unit shall have payloadType in the range of 36 to 44, inclusive.

When an MVC scalable nesting SEI message (payloadType equal to 37) or a view scalability information SEI message (payloadType equal to 38) or an opration point not present SEI message (payloadType equal to 43) is present in an SEI NAL unit, it shall be the only SEI message in the SEI NAL unit.

H.13.2.1 Parallel decoding information SEI message semantics

The parallel decoding information SEI message may be associated with any access unit. The information signalled in the SEI message applies to all the access units from the access unit the SEI message is associated with to the next access unit, in decoding order, containing an SEI message of the same type, exclusively, or to the end of the coded video sequence, whichever is earlier in decoding order.

Some view components for which the parallel decoding information is signalled in a parallel decoding information SEI message may be not present in the coded video sequence.

seq_parameter_set_id specifies a subset sequence parameter set that contains the inter-view dependency relationship information. The value of seq_parameter_set_id shall be equal to the value of seq_parameter_set_id in the picture parameter set referenced by a view component of the primary coded picture of the access unit containing the parallel decoding information SEI message. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

NOTE 1 – The inter-view dependency relationship is signalled in the sequence parameter set MVC extension, which is identical for all subset sequence parameter sets that may be activated during the decoding process for the coded video sequence.

pdi_init_delay_anchor_minus2_l0[i **][** j **]** specifies the unavailable reference area in the view component with view_id equal to anchor_ref_l0[i][j] that shall not be used for inter-view reference by the coded anchor view component with view_id equal to view_id[i], where anchor_ref_l0[i][j] and view_id[i] are both from the MVC sequence parameter set whose identifier is equal to the syntax element seq_parameter_set_id contained in the current SEI message. The unavailable reference area is a rectangular area with coordinates (0, (CurrMbAddr / PicWidthInMbs + pdi_init_delay_anchor_minus2_l0[i][j] + 2) * 16) as the top left corner and (PicWidthInSamples,PicHeightInSamples) as the bottom right corner. When decoding the coded view component with view_id equal to view_id[i], samples from the unavailable reference area from the view component with view_id equal to anchor_ref_l0[i][j] shall not be referred by the inter-view prediction process. The value of pdi_init_delay_anchor_minus2_l0[i][j] shall be in the range of 0 to PicHeightInMbs − 2, inclusive.

pdi_init_delay_anchor_minus2_l1[i **][** j **]** specifies the unavailable reference area in the view component with view_id equal to anchor_ref_l1[i][j] that shall not be used for inter-view reference by the coded anchor view component with view_id equal to view_id[i], where anchor_ref_lX[i][j] and view_id[i] are both from the MVC sequence parameter set whose identifier is equal to the syntax element seq_parameter_set_id contained in the current SEI message. The

unavailable reference area is a rectangular area with coordinates (0, (CurrMbAddr / PicWidthInMbs + pdi_init_delay_anchor_minus2_l1[i][j] + 2) * 16) as the top left corner and (PicWidthInSamples,PicHeightInSamples) as the bottom right corner. When decoding the coded view component with view_id equal to view_id[i], samples from the unavailable reference area from the view component with view_id equal to anchor_ref_l1[i][j] shall not be referred by the inter-view prediction process. The value of pdi_init_delay_anchor_minus2_l1[i][j] shall be in the range of 0 to PicHeightInMbs − 2, inclusive.

pdi_init_delay_non_anchor_minus2_l0[i **][** j **]** specifies the unavailable reference area in the view component with view_id equal to non_anchor_ref_l0[i][j] that shall not be used for inter-view reference by the coded non-anchor view component with view_id equal to view_id[i], where non_anchor_ref_l0[i][j] and view_id[i] are both from the MVC sequence parameter set whose identifier is equal to the syntax element seq_parameter_set_id contained in the current SEI message. The unavailable reference area is a rectangular area with coordinates (0, (CurrMbAddr / PicWidthInMbs + pdi_init_delay_non_anchor_minus2_l0[i][j] + 2) * 16) as the top left corner and (PicWidthInSamples,PicHeightInSamples) as the bottom right corner. When decoding the coded view component with view_id equal to view_id[i], samples from the unavailable reference area from the view component with view_id equal to non_anchor_ref_l0[i][j] shall not be referred by the inter-view prediction process. The value of pdi_init_delay_non_anchor_minus2_l0[i][j] shall be in the range of 0 to PicHeightInMbs − 2, inclusive.

pdi_init_delay_non_anchor_minus2_l1[i **][** j **]** specifies the unavailable reference area in the view component with view_id equal to non_anchor_ref_l1[i][j] that shall not be used for inter-view reference by the coded anchor view component with view_id equal to view_id[i], where non_anchor_ref_lX[i][j] and view_id[i] are both from the MVC sequence parameter set whose identifier is equal to the syntax element seq_parameter_set_id contained in the current SEI message. The unavailable reference area is a rectangular area with coordinates (0, (CurrMbAddr / PicWidthInMbs + pdi_init_delay_non_anchor_minus2_l1[i][j] + 2) * 16) as the top left corner and (PicWidthInSamples,PicHeightInSamples) as the bottom right corner. When decoding the coded view component with view_id equal to view_id[i], samples from the unavailable reference area from the view component with view_id equal to non_anchor_ref_l1[i][j] shall not be referred by the inter-view prediction process. The value of pdi_init_delay_non_anchor_minus2_l1[i][j] shall be in the range of 0 to PicHeightInMbs − 2, inclusive.

H.13.2.2 MVC scalable nesting SEI message semantics

An MVC nesting SEI message shall contain one and only one SEI message of payloadType less than or equal to 23, which is referred to as the nested SEI message. The scope to which the nested SEI message applies is indicated by the syntax elements operation_point_flag, all_view_components_in_au_flag, num_view_components_minus1, sei_view_id[i] for all i, num_view_components_op_minus1, sei_op_view_id[i] for all i, and sei_op_temporal_id.

Some view components to which the the nested SEI message applies may be not present in the access unit containing the MVC scalable nesting SEI message.

operation_point_flag[i] equal to 1 specifies that the nested SEI message applies to the current access unit when the associated operation point identified by sei_op_temporal_id and sei_op_view_id[i] for all i in the range of 0 to num_view_components_op_minus1, inclusive, is decoded. operation_point_flag[i] equal to 0 specifies that the nested SEI message applies to the view components identified by all_view_components_in_au_flag, num_view_components_minus1, and sei_view_id[i] for all i in the range of 0 to num_view_components_minus1, inclusive, regardless of which operation point is decoded.

If the nested SEI message has payloadType equal to 0 or 1, operation_point_flag[i] shall be equal to 1. Otherwise (the nested SEI message has payloadType not equal to 0 or 1),operation_point_flag[i] shall be equal to 0.

all_view_components_in_au_flag equal to 1 specifies that the nested SEI message applies to all view components of the access unit. all_view_components_in_au_flag equal to 0 specifies that the applicable scope of the nested SEI message is signalled by the syntax elements num_view_components_minus1 and sei_view_id[i] for all i in the range of 0 to num_view_components_minus1, inclusive.

num_view_components_minus1 plus 1 specifies the number of view components to which the nested SEI message applies when operation_point_flag is equal to 0 and all_view_components_in_au_flag is equal to 0. The value of num_view_components_minus1 shall be in the range of 0 to 1023, inclusive.

sei_view_id[i] specifies the view_id of the i-th view component to which the nested SEI message applies when operation_point_flag is equal to 0 and all_view_components_in_au_flag is equal to 0.

num_view_components_op_minus1 plus 1 specifies the number of view components of the operation point to which the nested SEI message applies when operation_point_flag is equal to 1. The value of num_view_components_op_minus1 shall be in the range of 0 to 1023, inclusive.

sei_op_view_id[i] specifies the view_id of the i-th view component to which the nested SEI message applies when operation_point_flag is equal to 1.

sei_op_temporal_id specifies the maximum temporal_id of the bitstream subset to which the nested SEI message applies when operation_point_flag is equal to 1.

sei_nesting_zero_bit is equal to 0.

H.13.2.3 View scalability information SEI message semantics

When present, this SEI message shall appear in an IDR access unit. The semantics of the message are valid for the current coded video sequence. A view scalability information SEI message contains view and scalability information for a subset of the opeation points in the coded video sequence. Each operation point is associated with an operation point identifier. The sub-bitstream for an operation point is referred to as the operation point representation or the representation of the operation point. Information such as bit rate and frame rate, among others, are signalled for the representations of the subset of the opeation points.

NOTE 1 – Any opreation point for which view and scalability information is signalled in a view scalability information SEI message (i.e. identified by a value of operation_point_id[i]) must be present in the coded video sequence. When an application keeps a view scalability information SEI message in a sub-bitstream extracted according to the process specified in subclause H.8.5.3, and after the extraction any opreation point for which view and scalability information is signalled in the orignal SEI message becomes not present in the coded video sequence, the application must change the content of the view scalability information SEI message to fullfil the condition stated by the first sentence in this note.

num_operation_point_minus1 plus 1 specifies the number of operation points that are present in the coded video sequence and for which the view scalability information is signalled by the following syntax elements. The value of num_operation_point_minus1 shall be in the range of 0 to 65535, inclusive.

The bitstream subset corresponding to an operation point is defined as the operation point representation or the representation of the operation point. The representation of the operation point identifed by operation_point_id[i] is the output of the sub-bitstream extraction process specified in subclause H.8.5.3 with tIdTarget equal to temporal_id[i] and viewIdTargetList consisting of view_id[i][j] for all j in the range of 0 to num_target_output_views_minus1[i], inclusive, as the inputs.

operation_point_id[i **]** specifies the identifier of the operation point. Each operation point is associated with a unique operation point identifier. The value of operation_point_id[i] shall be in the range of 0 to 65535, inclusive.

In the following semantics in this subclause, the operation point with identifier equal to operation_point_id[i] is referred to as the current operation point.

priority_id[i **]** and **temporal_id[** i **]** specify the maximum value of priority_id and temporal_id, respectively, of the NAL units in the represntaiton of the current operation point.

num_target_output_views_minus1[i **]** plus 1 specifies the number of target output views for the current operation point. The value of num_target_output_views_minus1[i]shall be in the range of 0 to 1023, inclusive.

view_id[i **][** j **]** specifies the identifier of the j-th target output view for the current operation point. The value of view_id[i][j] shall be in the range of 0 to 1023, inclusive.

profile_level_info_present_flag[i **]** equal to 1 specifies that the profile and level information for the representation of the current operation pointis present in the SEI message. profile_level_info_present_flag[i] equal to 0 specifies that the profile and level information for the current operation point is not present in the SEI message.

bitrate_info_present_flag[i **]** equal to 1 specifies that the bitrate information for the current operation point is present in the SEI message. bitrate_info_present_flag[i] equal to 0 specifies that the bitrate information for the current operation point is not present in the SEI message.

frm_rate_info_present_flag[i **]** equal to 1 specifies that the frame rate information for the current operation point is present in the SEI message. frm_rate_info_present_flag[i] equal to 0 specifies that the frame rate information for the current operation point is not present in the SEI message.

view_dependency_info_present_flag[i **]** equal to 1 specifies that information on the views the target output view of the current operation point directly depends on is present in the SEI message. View A is directly dependent on view point B if there is at least one view component of view A using a view component of view B for inter-view prediction reference. view_dependency_info_present_flag[i] equal to 0 specifies that view_dependency_info_src_op_id[i] is present in the SEI message. When not present, view_dependency_info_present_flag[i] shall be inferred to be equal to 0.

NOTE 2 – The inter-view dependency relationship signalled in sequence parameter set MVC extension is an upper limit, in the sense that whenever view A may depend on view B at any access unit, it is specified as view A depends on view B. Therefore, the dependency relationship is indicated by sequence parameter set MVC extension when view A depends on view B at only one of all access units in the coded video sequence, or even when view A actually does not depend on view B at any access unit but when generatiing the sequence parameter set MVC extension the encoder thought view A might depend on view B. The dependency relationship signalled here can be more refined. For example, when view A depends on view B at accesss units with temproal_id equal to 0 but not at other access units, this can be indicated through the view dependency information signalled in this SEI message for operation points with view A as the target output view and with different maximum values of temporal_id.

parameter_sets_info_present_flag[i **]** equal to 1 specifies that the values of seq_parameter_set_id of the sequence parameter sets and subset sequence parameter sets and the values of pic_parameter_set_id of the picture parameter sets that are referred to by the VCL NAL units of the representation of the current operation point are present in the SEI message. parameter_sets_info_present_flag[i] equal to 0 specifies that parameter_sets_info_src_op_id[i] is present in the SEI message.

bitstream_restriction_info_present_flag[i **]** equal to 1 specifies that the bitstream restriction information for the representation of the current operation point is present in the SEI message. bitstream_restriction_info_present_flag[i] equal to 0 specifies that the bitstream restriction information for the representation of the current operation point is not present in the SEI message.

op_profile_level_idc[i **]** specifies the profile and level compliancy of the representation of the current operation point. op_profile_level_idc[i] is the exact copy of the three bytes comprised of profile_idc, constraint_set0_flag, constraint_set1_flag, constraint_set2_flag, constraint_set3_flag, constraint_set4_flag, reserved_zero_3bits and level_idc, if these syntax elements were used to specify the profile and level compliancy of the representation of the current operation point as specified in Annexes A and H.

avg_bitrate[i **]** specifies the average bit rate of the representation of the current operation point. The average bit rate for the representation of the current operation point in bits per second is given by BitRateBPS(avg_bitrate[i]) with the function BitRateBPS( ) being specified by the following eqation.

$$\text{BitRateBPS}(x) = (x \,\&\, (214 - 1)) * 10(2 + (x >> 14))$$

All NAL units of the representation of the current operation point are taken into account in the calculation. The average bit rate is derived according to the access unit removal time specified in Annex C of this Recommendation | International Standard. In the following, bTotal is the number of bits in all NAL units of the representation of the current operation point in the current coded video sequence. t_1 is the removal time (in seconds) of the current access unit, and t_2 is the removal time (in seconds) of the last access unit (in decoding order) of the current coded video sequence.

With x specifying the value of avg_bitrate[i], the following appplies.

– If t_1 is not equal to t_2, the following condition shall be true.

$$(x \,\&\, (2^{14} - 1)) = = \text{Round}(\text{bTotal} \div ((t_2 - t_1) * 10^{(2 + (x >> 14))}))$$

– Otherwise (t_1 is equal to t_2), the following condition shall be true.

$$(x \,\&\, (2^{14} - 1)) = = 0$$

max_bitrate[i **]** specifies the maximum bit rate of the representation of the current operation point, given by BitRateBPS(max_bitrate_layer_representation[i]), in bits per second, with the function BitRateBPS( ) being specified in above. The maximum bit rate of the representation of the current operation point is calculated based on a time window specified by max_bitrate_calc_window[i].

max_bitrate_calc_window[i **]** specifies the length of the time window, in units of 1/100 second, based on which max_bitrate[i] is calculated.

constant_frm_rate_idc[i **]** specifies whether the frame rate of the representation of the current operation point is constant. If the value of avg_frm_rate as specified in below is constant whichever a temporal section of the operation point representation is used for the calculation, then the frame rate is constant, otherwise the frame rate is non-constant. constant_frm_rate_idc[i] equal to 0 specifies that the frame rate is not constant, constant_frm_rate_idc[i] equal to 1 specifies that the frame rate is constant, and constant_frm_rate_idc[i] equal to 2 specifies that the frame rate may be or may be not constant. The value of constant_frm_rate_idc[i] shall be in the range of 0 to 2, inclusive.

avg_frm_rate[i **]** specifies the average frame rate, in units of frames per 256 seconds, of the representation of the current operation point. The semantics of avg_frm_rate[i] is identical to the semantics of average_frame_rate in sub-sequence layer characteristics SEI message when accurate_statistics_flag is equal to 1, except that herein the set of NAL units in the range of sub-sequence layers is replaced by the set of NAL units of the representation of the current operation

point.

num_directly_dependent_views[i **]** specifies the number of views that the target output view of the current operation point is directly dependent on within the representation of the current operation point. The value of num_directly_dependent_views[i] shall be in the range of 0 to 16, inclusive.

directly_dependent_view_id[i **][** j **]** specifies the view_id of the j-th view that the target output view of the current operation point is directly dependent on within the representation of the current operation point. The value of directly_dependent_view_id[i][j] shall be in the range of 0 to 1023, inclusive.

view_dependency_info_src_op_id[i **]** specifies that the views the target output view of the current operation point directly depends on within the representation of the current operation popint are the same as the views the target output view of the operation point with identifier equal to view_dependency_info_src_op_id[i] directly depends on within the representation of the operation point with identifier equal to view_dependency_info_src_op_id[i], if view_dependency_info_src_op_id[i] is not equal to operation_point_id[i]. Otherwise (view_dependency_info_src_op_id[i] is equal to operation_point_id[i]), informaiton on the views the target output view of the current operation point directly depends on is not present in the SEI message. The value of view_dependency_info_src_op_id[i] shall be in the range of 0 to 65535, inclusive.

num_seq_parameter_set_minus1[i **]** plus 1 specifies the number of different sequence parameter sets that are referred to by the VCL NAL units of the representation of the current operation point. The value of num_seq_parameter_set_minus1[i] shall be in the range of 0 to 31, inclusive.

seq_parameter_set_id_delta[i **][** j **]** specifies the smallest value of the seq_parameter_set_id of all sequence parameter sets required for decoding the representation of the current operation point, if j is equal to 0. Otherwise (j is greater than 0), seq_parameter_set_id_delta[i][j] specifies the difference between the value of the seq_parameter_set_id of the j-th required sequence parameter set and the value of the seq_parameter_set_id of the (j-1)-th required sequence parameter set for decoding the representation of the current operation point. The sequence parameter sets are logically ordered in ascending order of the value of seq_parameter_set_id. The value of seq_parameter_set_id_delta[i][j] shall be in the range of 0 to 31, inclusive.

num_subset_seq_parameter_set_minus1[i **]** plus 1 specifies the number of different subset sequence parameter sets that are referred to by the VCL NAL units of the representation of the current operation point. The value of num_subset_seq_parameter_set_minus1[i] shall be in the range of 0 to 31, inclusive.

subset_seq_parameter_set_id_delta[i **][** j **]** specifies the smallest value of the seq_parameter_set_id of all subset sequence parameter sets required for decoding the representation of the current operation point, if j is equal to 0. Otherwise (j is greater than 0), subset_seq_parameter_set_id_delta[i][j] specifies the difference between the value of the seq_parameter_set_id of the j-th required subset sequence parameter set and the value of the seq_parameter_set_id of the (j-1)-th required subset sequence parameter set for decoding the representation of the current operation point. The subset sequence parameter sets are logically ordered in ascending order of the value of seq_parameter_set_id. The value of subset_seq_parameter_set_id_delta[i][j] shall be in the range of 0 to 31, inclusive.

num_pic_parameter_set_minus1[i **]** plus 1 specifies the number of different picture parameter sets that are referred to by the VCL NAL units of the representation of the current operation point. The value of num_pic_parameter_set_minus1[i] shall be in the range of 0 to 255, inclusive.

pic_parameter_set_id_delta[i **][** j **]** specifies the smallest value of the pic_parameter_set_id of all picture parameter sets required for decoding the representation of the current operation point, if j is equal to 0. Otherwise (j is greater than 0), pic_parameter_set_id_delta[i][j] specifies the difference between the value of the pic_parameter_set_id of the j-th required picture parameter set and the value of the pic_parameter_set_id of the (j-1)-th required picture parameter set for decoding the representation of the current operation point. The picture parameter sets are logically ordered in ascending order of the value of pic_parameter_set_id. The value of pic_parameter_set_id_delta[i][j] shall be in the range of 0 to 255, inclusive.

parameter_sets_info_src_op_id[i **]** specifies that the values of seq_parameter_set_id of the sequence parameter sets and subset sequence parameter sets and the values of pic_parameter_set_id of the picture parameter sets that are referred to by the VCL NAL units of the representation of the current opeation point are the same as those for the representation of the operation point with identifier equal to parameter_sets_info_src_op_id[i], if parameter_sets_info_src_op_id[i] is not equal to operation_point_id[i]. Otherwise (parameter_sets_info_src_op_id[i] is equal to operation_point_id[i]), parameter_sets_info_src_op_id[i] specifies that the values of seq_parameter_set_id of the sequence parameter sets and subset sequence parameter sets and the values of pic_parameter_set_id of the picture parameter sets that are referred to by the VCL NAL units of the representation of the current operation point are not present in the SEI message. The value of parameter_sets_info_src_op_id[i] shall be in the range of 0 to 65535, inclusive.

motion_vectors_over_pic_boundaries_flag[i **]** specifies the value of motion_vectors_over_pic_boundaries_flag, as specified in subclause E.2.1, for the current operation point representation. When the motion_vectors_over_pic_boundaries_flag[i] syntax element is not present, motion_vectors_over_pic_boundaries_flag value for the current operation point representation shall be inferred to be equal to 1.

max_bytes_per_pic_denom[i **]** specifies the max_bytes_per_pic_denom value, as specified in subclause E.2.1, for the current operation point representation. When the max_bytes_per_pic_denom[i] syntax element is not present, the value of max_bytes_per_pic_denom for the current operation point representation shall be inferred to be equal to 2. The value of max_bytes_per_pic_denom[i] shall be in the range of 0 to 16, inclusive.

max_bits_per_mb_denom[i **]** specifies the max_bits_per_mb_denom value, as specified in subclause E.2.1, for the current operation point representation. When the max_bits_per_mb_denom[i] is not present, the value of max_bits_per_mb_denom for the current operation point representation shall be inferred to be equal to 1. The value of max_bits_per_mb_denom[i] shall be in the range of 0 to 16, inclusive.

log2_max_mv_length_horizontal[i **]** and **log2_max_mv_length_vertical[** i **]** specify the values of log2_max_mv_length_horizontal and log2_max_mv_length_vertical, as specified in subclause E.2.1, for the current operation point representation. When log2_max_mv_length_horizontal[i] is not present, the values of log2_max_mv_length_horizontal and log2_max_mv_length_vertical for the current operation point representation shall be inferred to be equal to 16. The value of log2_max_mv_length_horizontal[i] shall be in the range of 0 to 16, inclusive. The value of log2_max_mv_length_vertical[i] shall be in the range of 0 to 16, inclusive.

NOTE 3 – The maximum absolute value of a decoded vertical or horizontal motion vector component is also constrained by profile and level limits as specified in Annex A or Subclause H.10.2.

num_reorder_frames[i **]** specifies the value of num_reorder_frames, as specified in subclause E.2.1, for the current operation point representation. The value of num_reorder_frames[i] shall be in the range of 0 to 16, inclusive. When the num_reorder_frames[i] syntax element is not present, the value of num_reorder_frames for the current operation point representation shall be inferred to be equal to 16.

max_dec_frame_buffering[i **]** specifies the value of max_dec_frame_buffering, as specified in subclause E.2.1, for the current operation point representation. The value of max_dec_frame_buffering[i] shall be in the range of 0 to MaxDpbFrames (as specified in subclause A.3.1, A.3.2, or H.10.2), inclusive. When the max_dec_frame_buffering[i] syntax element is not present, the value of max_dec_frame_buffering for the current operation point representation shall be inferred to be equal to MaxDpbFrames.

H.13.2.4 Multiview scene information SEI message semantics

The multiview scene information SEI message indicates the maximum disparity among multiple view components in an access unit. The maximum disparity could be used for processing the decoded view components prior to rendering on a 3D display. When present, the multiview scene information SEI message shall be associated with an IDR access unit. The information signaled in the SEI message applies to the coded video sequence.

The actual maximum disparity value may be smaller than the one signalled in the multiview scene information SEI message, due to that some views in the coded video sequence may have been removed from the original bitstream to produce an extracted sub-bitstream according to the process specified in subclause H.8.5.3.

max_disparity specifies the maximum disparity in units of integer pixel resolution between spatially adjacent view components among the total set of view components in an access unit. The value of max_disparity shall be in the range of 0 to 1023, inclusive.

NOTE 1 – The maximum disparity depends on the baseline distance between spatially adjacent views and the spatial resolution of each view. Therefore, if either the number of views or spatial resolution is changed, the maximum disparity should also be changed accordingly.

H.13.2.5 Multiview acquisition information SEI message semantics

The multiview acquisition information SEI message specifies various parameters of the acquisition environment. Specifically, intrinsic and extrinsic camera parameters are specified. These parameters could be used for processing the decoded view components prior to rendering on a 3D display. When present, the multiview acquisition information SEI message shall be associated with an IDR access unit. The information signaled in the SEI message applies to the coded video sequence.

Some of the views for which the multiview acquisition information is included in a multiview acquisition information SEI message may be not present in the coded video sequence.

The extrinsic camera parameters shall be specified according to a right-handed coordinate system, where the upper left

corner of the image is the origin, i.e., the (0, 0) coordinate, with all other corners of the image having non-negative coordinates. With these specifications, a 3-dimensional world point, wp=[x y z] is mapped into a 2 - dimensional camera point, cp = s * [u v 1], for the i-th camera according to:

$$s * cp(i) = A(i) * R^{-1}(i) * [wp - t(i)] \quad \text{(H-9)}$$

where A(i) denotes the intrinsic camera parameters, $R^{-1}(i)$ denotes the inverse of the rotation matrix R(i), t(i) denotes the translation vector, and s is an arbitrary scaling chosen to make the third coordinate of cp equal to one.

num_views_minus1 shall be equal to the value of the syntax element num_views_minus1 in the active MVC sequence parameter set for the coded video sequence. The value of number_of_view_minus1 shall be in the range of 0 to 1023, inclusive.

intrinsic_param_flag equal to 1 indicates the presence of intrinsic camera parameters. intrinsic_param_flag equal to 0 indicates the absence of intrinsic camera parameters.

extrinsic_param_flag equal to 1 indicates the presence of extrinsic camera parameters. extrinsic_param_flag equal to 0 indicates the absence of extrinsic camera parameters.

intrinsic_params_equal equal to 1 indicates that the intrinsic camera parameters are equal for all cameras and only one set of intrinsic camera parameters are present. intrinsic_params_equal equal to 0 indicates that the intrinsic camera parameters are different for each camera and that a set of intrinsic camera parameters are present for each camera.

prec_focal_length specifies the exponent of the maximum allowable truncation error for focal_length_x[i] and focal_length_y[i] as given by $2^{-prec\_focal\_length}$. The value of prec_focal_length shall be in the range of 0 to 31, inclusive.

prec_principal point specifies the exponent of the maximum allowable truncation error for principal_point_x[i] and principal_point_y[i] as given by $2^{-prec\_principal\_point}$. The value of prec_principal point shall be in the range of 0 to 31, inclusive.

prec_skew_factor specifies the exponent of the maximum allowable truncation error for skew factor as given by $2^{-prec\_skew\_factor}$. The value of prec_skew_factor shall be in the range of 0 to 31, inclusive.

sign_focal_length_x[i **]** equal to 0 indicates that the sign of the focal length of the i-th camera in the horizontal direction is positive. sign_focal_length_x[i] equal to 1 indicates the sign is negative.

exponent_focal_length_x[i **]** specifies the exponent part of the focal length of the i-th camera in the horizontal direction. The value of exponent_focal_length_x[i] shall be in the range of 0 to 62, inclusive. The value of 63 is reserved for future use by ITU-T | ISO/IEC.

mantissa_focal_length_x[i **]** specifies the mantissa part of the focal length of the i-th camera in the horizontal direction. The size of the mantissa_focal_length_x[i] syntax element is variable and determined as follows.

– If exponent_focal_length_x[i] = = 0, then the size is max(0, prec_focal_length - 30).

– Otherwise (if 0 < exponent_focal_length_x[i] < 63), then the size is max(0, exponent_focal_length_x[i] + prec_focal_length - 31).

sign_focal_length_y[i **]** equal to 0 indicates that the sign of the focal length of the i-th camera in the vertical direction is positive. sign_focal_length_y[i] equal to 1 indicates the sign is negative.

exponent_focal_length_y[i **]** specifies the exponent part of the focal length of the i-th camera in the vertical direction. The value of exponent_focal_length_y[i] shall be in the range of 0 to 62, inclusive. The value of 63 is reserved for future use by ITU-T | ISO/IEC.

mantissa_focal_length_y[i **]** specifies the mantissa part of the focal length of the i-th camera in the vertical direction. The size of the mantissa_focal_length_y[i] syntax element is variable and determined as follows.

– If exponent_focal_length_y[i] = = 0, then the size is max(0, prec_focal_length - 30)

– Otherwise (if 0 < exponent_focal_length_y[i] < 63), then the size is max(0, exponent_focal_length_y[i] + prec_focal_length - 31)

sign_principal_point_x[i **]** equal to 0 indicates that the sign of the principal point of the i-th camera in the horizontal direction is positive. sign_principal_point_x[i] equal to 1 indicates the sign is negative.

exponent_principal_point_x[i **]** specifies the exponent part of the principal point of the i-th camera in the horizontal

direction. The value of exponent_principal_point_x[i] shall be in the range of 0 to 62, inclusive. The value of 63 is reserved for future use by ITU-T | ISO/IEC.

mantissa_principal_point_x[i **]** specifies the mantissa part of the principal point of the i-th camera in the horizontal direction. The size of the mantissa_principal_point_x[i] syntax element is variable and determined as follows.

– If exponent_principal_point_x[i] = = 0, then the size is max(0, prec_principal_point - 30)

– Otherwise (if 0 < exponent_principal_point_x[i] < 63), then the size is max(0, exponent_principal_point_x[i] + prec_principal_point - 31)

sign_principal_point_y[i **]** equal to 0 indicates that the sign of the principal point of the i-th camera in the vertical direction is positive. sign_principal_point_y[i] equal to 1 indicates the sign is negative.

exponent_principal_point_y[i **]** specifies the exponent part of the principal point of the i-th camera in the vertical direction. The value of exponent_principal_point_y[i] shall be in the range of 0 to 62, inclusive. The value of 63 is reserved for future use by ITU-T | ISO/IEC.

mantissa_principal_point_y[i **]** specifies the mantissa part of the principal point of the i-th camera in the vertical direction. The size of the mantissa_principal_point_y[i] syntax element is variable and determined as follows.

– If exponent_principal_point_y[i] = = 0, then the size is max(0, prec_principal_point - 30)

– Otherwise (if 0 < exponent_principal_point_y[i] < 63), then the size is max(0, exponent_principal_point_y[i] + prec_principal_point - 31)

sign_skew_factor[i **]** equal to 0 indicates that the sign of the skew factor of the i-th camera is positive. sign_skew_factor[i] equal to 1 indicates the sign is negative.

exponent_skew_factor[i **]** specifies the exponent part of the skew factor of the i-th camera. The value of exponent_skew_factor[i] shall be in the range of 0 to 62, inclusive. The value of 63 is reserved for future use by ITU-T | ISO/IEC.

mantissa_skew_factor[i **]** specifies the mantissa part of the skew factor of the i-th camera. The size of the mantissa_skew_factor[i] syntax element is variable and determined as follows.

– If exponent_skew_factor[i] = = 0, then the size is max(0, prec_skew_factor - 30)

– Otherwise (if 0 < exponent_skew_factor[i] < 63), then the size is max(0, exponent_skew_factor[i] + prec_skew_factor - 31)

NOTE 1 – The intrinsic matrix A(i) for i-th camera is represented as follows:

$$\begin{bmatrix} \text{focal\_length\_x}[\,i\,] & \text{skew\_factor}[\,i\,] & \text{principal\_point\_x}[\,i\,] \\ 0 & \text{focal\_length\_y}[\,i\,] & \text{principal\_point\_y}[\,i\,] \\ 0 & 0 & 1 \end{bmatrix}$$

prec_rotation_param specifies the exponent of the maximum allowable truncation error for r[i][j][k] as given by $2^{-prec\_rotation\_param}$. The value of prec_rotation_param shall be in the range of 0 to 31, inclusive.

prec_translation_param specifies the exponent of the maximum allowable truncation error for t[i][j] as given by $2^{-prec\_translation\_param}$. The value of prec_ translation_param shall be in the range of 0 to 31, inclusive.

sign_r[i **][** j **][** k **]** equal to 0 indicates that the sign of (j, k) component of the rotation matrix for the i-th camera is positive. sign_r[i][j][k] equal to 1 indicates the sign is negative.

exponent_r[i **][** j **][** k **]** specifies the exponent part of (j, k) component of the rotation matrix for the i-th camera. The value of exponent_r[i][j][k] shall be in the range of 0 to 62, inclusive. The value of 63 is reserved for future use by ITU-T | ISO/IEC.

mantissa_r[i **][** j **][** k **]** specifies the mantissa part of (j, k) component of the rotation matrix for the i-th camera. The size of the mantissa_r[i][j][k] syntax element is variable and determined as follows.

– If exponent_r[i] = = 0, then the size is max(0, prec_rotation_param - 30)

– Otherwise (if 0 < exponent_r[i] < 63), then the size is max(0, exponent_r[i] + prec_rotation_param - 31)

NOTE 2 – The rotation matrix R(i) for i-th camera is represented as follows.

$$\begin{bmatrix} r[\,i\,][\,0\,][\,0\,] & r[\,i\,][\,0\,][\,1\,] & r[\,i\,][\,0\,][\,2\,] \\ r[\,i\,][\,1\,][\,0\,] & r[\,i\,][\,1\,][\,1\,] & r[\,i\,][\,1\,][\,2\,] \\ r[\,i\,][\,2\,][\,0\,] & r[\,i\,][\,2\,][\,1\,] & r[\,i\,][\,2\,][\,2\,] \end{bmatrix}$$

sign_t[i **][** j **]** equal to 0 indicates that the sign of the j-th component of the translation vector for the i-th camera is positive. sign_t[i][j] equal to 1 indicates the sign is negative.

exponent_t[i **][** j **]** specifies the exponent part of the j-th component of the translation vector for the i-th camera. The value of exponent_t[i][j] shall be in the range of0 to 62, inclusive. The value of 63 is reserved for future use by ITU-T | ISO/IEC.

mantissa_t[i **][** j **]** specifies the mantissa part of the j-th component of the translation vector for the i-th camera. The size of the mantissa_t[i][j] syntax element is variable and determined as follows.

– If exponent_t[i] = = 0, then the size is max(0, prec_translation_param - 30)

– Otherwise (if 0 < exponent_t[i] < 63), then the size is max(0, exponent_t[i] + prec_translation_param - 31)

NOTE 3 – The translation vector t(i) for i-th camera is represented as follows:

$$\begin{bmatrix} t[\,i\,][\,0\,] \\ t[\,i\,][\,1\,] \\ t[\,i\,][\,2\,] \end{bmatrix}$$

NOTE 4 – The components of the intrinsic and rotation matrices as well as the translation vector are obtained as follows according to the decoded syntax elements and the IEC 60559 specification:

If 0 < E < 63, then $\boldsymbol{X} = (-1)^{s} \cdot 2^{E-31} \cdot (1.\boldsymbol{M})$.

If E = 0 and M is non-zero, then $\boldsymbol{X} = (-1)^{s} \cdot 2^{-30} \cdot (0.\boldsymbol{M})$.

If E = 0 and M = 0, then $\boldsymbol{X} = (-1)^{s} \cdot 0$,

where X is the variable to be calculated, s , N and E correspond to the associated sign, exponent and mantissa syntax elements for each variable to be calculated, and M = bin2float(N) with 0 <= M < 1. The association between each camera parameter variable and corresponding syntax elements is given by Table H-2. NOTE 5 provides psuedo-code for the bin2float( ) function that converts a binary representation of a fractional number into a corresponding floating-point number.

Table H-2 – Association between camera parameter variables and syntax elements.

X	s	E	N
focal_length_x[i **]**	sign_focal_length_x[i]	exponent_focal_length_x[i]	mantissa_focal_length_x[i]
focal_length_y[i **]**	sign_focal_length_y[i]	exponent_focal_length_y[i]	mantissa_focal_length_y[i]
principal_point_x[i **]**	sign_principal_point_x[i]	exponent_principal_point_x[i]	mantissa_principal_point_x[i]
principal_point_y[i **]**	sign_principal_point_y[i]	exponent_principal_point_y[i]	mantissa_principal_point_y[i]
skew_factor[i **]**	sign_skew_factor[i]	exponent_skew_factor[i]	mantissa_skew_factor[i]
r[i **][** j **][** k **]**	sign_r[i][j][k]	exponent_r[i][j][k]	mantissa _r[i][j][k]
t[i **][** j **]**	sign_t[i][j]	exponent_t[i][j]	mantissa_t[i][j]

NOTE 5 – To convert a binary representation of a fractional number N (0 <= N < 1) into the corresponding floating-point number M, the psuedo-code of a function M = bin2float(N) is given as follows:

```
float M = 0;
float factor = 2^(-v); /* v is the length of the mantissa */
for ( i = 0; i < v; i++) {
      M = M + factor * (N>> i) & 0x01;
      factor = factor * 2;
}
```

H.13.2.6 Non-required view component SEI message semantics

This SEI message indicates non-required view components within the associated access unit. A view component is a non-required view component for a target view component if it is not needed for decoding the target view component and

subsequent view components with the same view_id in decoding order within the coded video sequence.

Some of the view components indicated by view_order_index[i] or index_delta_minus1[i][j] may be not present in the assoicated access unit.

num_info_entries_minus1 plus 1 specifies the number of target view components for which non-required view components are indicated. The value of num_info_entries_minus1 shall be in the range of 0 to num_views_minus1 - 1, inclusive.

view_order_index[i **]** specifies the view order index of the i-th target view component for which non-required view components are indicated. The i-th target view component has view_id equal to view_id[view_order_index [i]]. The value of view_order_index[i] shall be in the range of 1 to num_views_minus1, inclusive.

num_non_required_view_components_minus1[i] plus 1 specifies the number of non-required view components for the i-th target view component. The value of num_non_required_view_components_minus1[i] shall be in the range of 0 to view_order_index[i] – 1, inclusive.

index_delta_minus1[i **][** j **]** plus 1 specifies the difference between the view order index of the i-th target view component and the view order index of the j-th non-required view component for the i-th target view component. The view order index of the j-th non-required view component for the i-th target view component is view_order_index[i] – index_delta_minus1[i][j] – 1. The value of index_delta_minus1[i][j] shall be in the range of 0 to view_order_index[i] – 1, inclusive.

H.13.2.7 View dependency change SEI message semantics

This SEI message indicates that the view dependency information changes starting with the current access unit containing the SEI message and is always interpreted with respect to the active MVC sequence parameter set. When present, the view dependency change SEI message applies to the target access unit set that consists of the current access unit and all the subsequent access units, in decoding order, until the next view dependency change SEI message or the end of the coded video sequence, whichever is earlier in decoding order.

If, according to the view dependency information indicated in the active MVC sequence parameter set, view component A does not directly or indirectly depend on view component B and vice versa, the view dependency change SEI message shall not specify view dependency relationship between view components A and B.

NOTE 1 – The dependent views for any view are always a subset of those indicated by the active MVC sequence parameter set.

NOTE 2 – View dependency change SEI messages do not have a cumulative effect.

Some of the views indicated by the following syntax elements may be not present in the target access unit set.

seq_parameter_set_id specifies a subset sequence parameter set that contains the inter-view dependency relationship information. The value of seq_parameter_set_id shall be equal to the value of seq_parameter_set_id in the picture parameter set referenced by a view component of the primary coded picture of the access unit containing the view dependency change SEI message. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

anchor_update_flag equal to 1 indicates that there are updates for the dependencies for anchor view components comparing to the dependencies defined in the active MVC sequence parameter set. anchor_update_flag equal to 0 indicates that there is no change for the dependencies for anchor view components comparing to the dependencies defined in the active MVC sequence parameter set.

non_anchor_update_flag equal to 1 indicates that there are updates for the dependencies for non-anchor view components comparing to the dependencies defined in active MVC sequence parameter set. non_anchor_update_flag equal to 0 indicates that there is no change for the dependencies for non-anchor view components comparing to the dependencies defined in the active MVC sequence parameter set.

anchor_ref_l0_flag[i **][** j **]** equal to 0 indicates that the j-th inter-view prediction reference in the initialised RefPicList0 for any anchor view component with view order index equal to i will not be present in the final RefPicList0 after reference pictue list modification for the anchor view compoent. anchor_ref_l0_flag[i][j] equal to 1 indicates that the j-th inter-view prediction reference in the initialised RefPicList0 for at least one anchor view component with view order index equal to i will be present in the final RefPicList0 after reference pictue list modification for the anchor view compoent.

anchor_ref_l1_flag[i **][** j **]** equal to 0 indicates that the j-th inter-view prediction reference in the initialised RefPicList1 for any anchor view component with view order index equal to i will not be present in the final RefPicList1 after reference pictue list modification for the anchor view compoent. anchor_ref_l1_flag[i][j] equal to 1 indicates that the j-th inter-view prediction reference in the initialised RefPicList1 for at least one anchor view component with view order

index equal to i will be present in the final RefPicList1 after reference pictue list modification for the anchor view compoent.

non_anchor_ref_l0_flag[i **][** j **]** equal to 0 indicates that the j-th inter-view prediction reference in the initialised RefPicList0 for any non-anchor view component with view order index equal to i will not be present in the final RefPicList0 after reference pictue list modification for the non-anchor view compoent. non-anchor_ref_l0_flag[i][j] equal to 1 indicates that the j-th inter-view prediction reference in the initialised RefPicList0 for at least one non-anchor view component with view order index equal to i will be present in the final RefPicList0 after reference pictue list modification for the non-anchor view compoent.

non_anchor_ref_l1_flag[i **][** j **]** equal to 0 indicates that the j-th inter-view prediction reference in the initialised RefPicList1 for any non-anchor view component with view order index equal to i will not be present in the final RefPicList1 after reference pictue list modification for the non-anchor view compoent. non-anchor_ref_l1_flag[i][j] equal to 1 indicates that the j-th inter-view prediction reference in the initialised RefPicList1 for at least one non-anchor view component with view order index equal to i will be present in the final RefPicList1 after reference pictue list modification for the non-anchor view compoent.

H.13.2.8 Operation point not present SEI message semantics

This SEI message indicates operation points that are not present in the bitstream starting with the current access unit, and is interpreted with respect to the previous view scalability information SEI message in decoding order. The message remains effective until the next SEI message of the same type or the end of the coded video sequence, whichever is earlier in decoding order.

NOTE 1– Operation point not present SEI messages do not have a cumulative effect.

NOTE 2 – Any opreation point identified by a value of operation_point_id[i] in the previous view scalability information SEI message, in decoding order, and not identified by a value of operation_point_not_present_id[k] must be present in the coded video sequence. Therefore, when an application keeps an operation point not present SEI message in a sub-bitstream extracted according to the process specified in subclause H.8.5.3, the application may need to change the content of the operation point not present SEI message according to the semantics.

num_operation_points specifies the number of operation points that are indicated not to be present by the SEI message. num_operation_points equal to 0 indicates that all operation points indicated by the view scalability information SEI message are present. The value of num_operation_points shall be in the range of 0 to the value of num_operation_points_minus1 in the previous view scalability information SEI message in decoding order, inclusive.

operation_point_not_present_id[k **]** identifies an operation point that is not present. operation_point_not_present_id[k] shall be equal to the value of one of the operation_point_id[i] syntax elements of the previous view scalability information SEI message in decoding order. The value of operation_point_not_present_id[k] shall be in the range of 0 to 65535, inclusive.

H.13.2.9 Base view temporal HRD SEI message semantics

When present, this SEI message shall be assoicated with an IDR access unit and applies to the coded video sequence. Some temporal subsets identifyed by temporal_id[i] may be not present in the coded video sequence.

num_of_temporal_layers_in_base_view_minus1 plus 1 specifies the number of temporal bitstream subsets in the coded video sequence for which the following syntax elements apply. The value of num_of_temporal_layers_in_base_view_minus1 shall be in the range of 0 to 7, inclusive.

temporal_id[i **]** specifies the temporal_id value of the i-th temporal bitstream subset.

Let the i-th bitstream subset for the coded video sequence that is obtained by invoking the sub-bitstream extraction process as specified in subclause H.8.5.3 with tIdTarget equal to temporal_id[i] as input.

timing_info_present_flag[i **]** specifies the value of timing_info_present_flag that applies for the i-th bitstream subset.

num_units_in_tick[i **]** specifies the value of num_units_in_tick that applies for the i-th bitstream subset. num_units_in_tick[i] shall be greater than 0.

time_scale[i **]** specifies the value of time_scale that applies for the i-th bitstream subset. time_scale[i] shall be greater than 0.

fixed_frame_rate_flag[i **]** specifies the value of fixed_frame_rate_flag that applies for the i-th bitstream subset.

nal_hrd_parameters_present_flag[i **]** specifies the value of nal_hrd_parameters_present_flag that applies for the i-th bitstream subset.

vcl_hrd_parameters_present_flag[i **]** specifies the value of vcl_hrd_parameters_present_flag that applies for the i-th bitstream subset.

low_delay_hrd_flag[i **]** specifies the value of low_delay_hrd_flag that applies for the i-th bitstream subset.

pic_struct_present_flag[i **]** specifies the value of pic_struct_present_flag that applies for the i-th bitstream subset.

H.14 MVC video usability information extension

The specifications in Annex E apply. Additionally, the following applies.

H.14.1 MVC VUI parameters extension syntax

mvc_vui_parameters_extension( ) {	C	Descriptor
num_ops_minus1	0	ue(v)
for(i = 0; i <= num_ops_minus1; i++) {		
temporal_id[i **]**	0	u(3)
num_target_output_views_minus1[i **]**	5	ue(v)
for(j = 0; j <= num_target_output_views_minus1[i]; j++)		
view_id[i **][** j **]**	5	ue(v)
timing_info_present_flag[i **]**	0	u(1)
if(timing_info_present_flag[i]) {		
num_units_in_tick[i **]**	0	u(32)
time_scale[i **]**	0	u(32)
fixed_frame_rate_flag[i **]**	0	u(1)
}		
nal_hrd_parameters_present_flag[i **]**	0	u(1)
if(nal_hrd_parameters_present_flag[i])		
hrd_parameters( )		
vcl_hrd_parameters_present_flag[i **]**	0	u(1)
if(vcl_hrd_parameters_present_flag[i])		
hrd_parameters( )		
if(nal_hrd_parameters_present_flag[i] || vcl_hrd_parameters_present_flag[i])		
low_delay_hrd_flag[i **]**	0	u(1)
pic_struct_present_flag[i **]**	0	u(1)
}		
}		

H.14.2 MVC VUI parameters extension semantics

The MVC VUI parameters extension specifies VUI parameters that apply to one or more operation points for the coded video sequence. All MVC VUI parameters extensions that are referred to by a coded video sequence shall be identical.

Some views identified by view_id[i][j] may be not present in the coded video sequence. Some temporal subsets identifyed by temporal_id[i] may be not present in the coded video sequence.

num_ops_minus1 plus 1 specifies the number of operation points for which timing information, NAL HRD parameters, VCL HRD parameters and the pic_struct_present_flag may be present. The value of num_ops_minus1 shall be in the range of 0 to 65535, inclusive..

temporal_id[i] indicates the maximum value of temporal_id for all VCL NAL units in the representation of the i-th operation point.

num_target_output_views_minus1[i **]** plus one specifies the number of target output views for the i-th operation point. The value of num_target_output_views_minus1[i] shall be in the range of 0 to 1023, inclusive.

view_id[i][j] indicates the j-th target output view in the i-th operation point. The value of view_id[i] shall be in the range of 0 to 1023, inclusive.

The following syntax elements apply to the coded video sequence that is obtained by the sub-bitstream extraction process as specified in subclause H.8.5.3 with tIdTarget equal to temporal_id[i] and viewIdTargetList containing view_id[i][j] for all j in the range of 0 to num_target_output_views_minus1[i], inclusive, as the inputs and the i-th sub-bitstream as the output.

timing_info_present_flag[i] specifies the timing_info_present_flag value of the i-th sub-bitstream.

num_units_in_tick[i] specifies the num_units_in_tick value of the i-th sub-bitstream.

time_scale[i] specifies the time_scale value of the i-th sub-bitstream.

fixed_frame_rate_flag[i] specifies the fixed_frame_rate_flag value of the i-th sub-bitstream.

nal_hrd_parameters_present_flag[i] specifies the nal_hrd_parameters_present_flag value of the i-th sub-bitstream.

vcl_hrd_parameters_present_flag[i] specifies the vcl_hrd_parameters_present_flag value of the i-th sub-bitstream.

low_delay_hrd_flag[i] specifies the low_delay_hrd_flag value of the i-th sub-bitstream.

pic_struct_present_flag[i] specifies the pic_struct_present_flag value of the i-th sub-bitstream.