DRAFT AMENDMENT ISO/IEC 14496-10:2012/DAM 2

ISO/IEC JTC **1** Secretariat: **ANSI**

Voting begins on **2012-10-09** Voting terminates on **2013-01-09**

Information technology — Coding of audio-visual objects —

Part 10: **Advanced Video Coding**

AMENDMENT 2

Technologies de l'information — Codage des objets audiovisuels —

Partie 10: Codage visuel avancé

AMENDEMENT 2

ICS 35.040

To expedite distribution, this document is circulated as received from the committee secretariat. ISO Central Secretariat work of editing and text composition will be undertaken at publication stage.

Pour accélérer la distribution, le présent document est distribué tel qu'il est parvenu du secrétariat du comité. Le travail de rédaction et de composition de texte sera effectué au Secrétariat central de l'ISO au stade de publication.

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 2 to ISO/IEC 14496-10:2012 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technolog — Coding of audio-visual objects — Part 10: Advanced Video Coding, AMENDMENT 2: MVC extensions for inclusion of depth maps

AMENDMENT 2

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

An extension of multiview video coding that also supports the inclusion of depth maps is specified in Annex I allowing the construction of bitstreams that represent multiple views with corresponding depth views. Similar to multiview video coding, bitstreams that include multiple depth 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. For view bitstream scalability, i.e. the presence of a sub-bitstream with fewer views than those included in 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 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 H specifies":

Annex I specifies MVC extensions for inclusion of depth maps, referred to as 3D Video Coding (3DVC). The reader is referred to Annex I for the entire decoding process for 3DVC, which is specified there with references being made to clauses 2-9 and Annexes A-E and Annex H. Subclause I.10 specifies one profile for 3DVC (Multiview and Depth).

In 7.3.1, replace the syntax table with:

nal_unit(NumBytesInNALunit) {	**C**	**Descriptor**
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)
NumBytesInRBSP = 0		
nalUnitHeaderBytes = 1		
if(nal_unit_type = = 14 || nal_unit_type = = 20 || nal_unit_type = = 21) {		
svc_extension_flag	All	u(1)
if(!svc_extension_flag || nal_unit_type = = 21))		
nal_unit_header_mvc_extension( ) /* specified in Annex H */	All	
else		
nal_unit_header_svc_extension( ) /* specified in Annex G */	All	
nalUnitHeaderBytes += 3		
}		
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {		
if(i + 2 < NumBytesInNALunit && next_bits(24) = = 0x000003) {		
rbsp_byte[NumBytesInRBSP++ **]**	All	b(8)
rbsp_byte[NumBytesInRBSP++ **]**	All	b(8)
i += 2		
emulation_prevention_three_byte /* equal to 0x03 */	All	f(8)
} else		
rbsp_byte[NumBytesInRBSP++ **]**	All	b(8)
}		
}		

In 7.3.2.1.3, replace the syntax table with:

subset_seq_parameter_set_rbsp( ) {	C	Descriptor
seq_parameter_set_data( )	0	
if(profile_idc = = 83 || profile_idc = = 86) {		
seq_parameter_set_svc_extension( ) /* specified in Annex G */	0	
svc_vui_parameters_present_flag	0	u(1)
if(svc_vui_parameters_present_flag = = 1)		
svc_vui_parameters_extension( ) /* specified in Annex G */	0	
} else if(profile_idc = = 118 || profile_idc = = 128) {		
bit_equal_to_one /* equal to 1 */	0	f(1)
seq_parameter_set_mvc_extension( ) /* specified in Annex H */	0	
mvc_vui_parameters_present_flag	0	u(1)
if(mvc_vui_parameters_present_flag = = 1)		
mvc_vui_parameters_extension( ) /* specified in Annex H */	0	
}		
if(profile_idc = = 138) {		
bit_equal_to_one /* equal to 1 */	0	f(1)
seq_parameter_set_mvc_extension( ) /* specified in Annex H */	0	
seq_parameter_set_3dvc_extension( )	0	
}		
additional_extension3_flag	0	u(1)
if(additional_extension3_flag = = 1)		
while(more_rbsp_data( ))		
additional_extension3_data_flag	0	u(1)
rbsp_trailing_bits( )	0	
}		

In 7.3.3, replace the syntax table with:

slice_header( ) {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
if(separate_colour_plane_flag = = 1)		
colour_plane_id	2	u(2)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(IdrPicFlag)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type = = 0) {		
pic_order_cnt_lsb	2	u(v)
if(bottom_field_pic_order_in_frame_present_flag && !field_pic_flag)		

<table>
<tr><td>delta_pic_order_cnt_bottom</td><td>2</td><td>se(v)</td></tr>
<tr><td>}</td><td></td><td></td></tr>
<tr><td>if(pic_order_cnt_type = = 1 && !delta_pic_order_always_zero_flag) {</td><td></td><td></td></tr>
<tr><td>delta_pic_order_cnt[0]</td><td>2</td><td>se(v)</td></tr>
<tr><td>if(bottom_field_pic_order_in_frame_present_flag && !field_pic_flag)</td><td></td><td></td></tr>
<tr><td>delta_pic_order_cnt[1]</td><td>2</td><td>se(v)</td></tr>
<tr><td>}</td><td></td><td></td></tr>
<tr><td>if(redundant_pic_cnt_present_flag)</td><td></td><td></td></tr>
<tr><td>redundant_pic_cnt</td><td>2</td><td>ue(v)</td></tr>
<tr><td>if(slice_type = = B)</td><td></td><td></td></tr>
<tr><td>direct_spatial_mv_pred_flag</td><td>2</td><td>u(1)</td></tr>
<tr><td>if(slice_type = = P || slice_type = = SP || slice_type = = B) {</td><td></td><td></td></tr>
<tr><td>num_ref_idx_active_override_flag</td><td>2</td><td>u(1)</td></tr>
<tr><td>if(num_ref_idx_active_override_flag) {</td><td></td><td></td></tr>
<tr><td>num_ref_idx_l0_active_minus1</td><td>2</td><td>ue(v)</td></tr>
<tr><td>if(slice_type = = B)</td><td></td><td></td></tr>
<tr><td>num_ref_idx_l1_active_minus1</td><td>2</td><td>ue(v)</td></tr>
<tr><td>}</td><td></td><td></td></tr>
<tr><td>}</td><td></td><td></td></tr>
<tr><td>if(nal_unit_type = = 20 || nal_unit_type = = 21)</td><td></td><td></td></tr>
<tr><td>ref_pic_list_mvc_modification( ) /* specified in Annex H */</td><td>2</td><td></td></tr>
<tr><td>else</td><td></td><td></td></tr>
<tr><td>ref_pic_list_modification( )</td><td>2</td><td></td></tr>
<tr><td>if((weighted_pred_flag && (slice_type = = P || slice_type = = SP)) ||
(weighted_bipred_idc = = 1 && slice_type = = B))</td><td></td><td></td></tr>
<tr><td>pred_weight_table( )</td><td>2</td><td></td></tr>
<tr><td>if(nal_ref_idc != 0)</td><td></td><td></td></tr>
<tr><td>dec_ref_pic_marking( )</td><td>2</td><td></td></tr>
<tr><td>if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)</td><td></td><td></td></tr>
<tr><td>cabac_init_idc</td><td>2</td><td>ue(v)</td></tr>
<tr><td>slice_qp_delta</td><td>2</td><td>se(v)</td></tr>
<tr><td>if(slice_type = = SP || slice_type = = SI) {</td><td></td><td></td></tr>
<tr><td>if(slice_type = = SP)</td><td></td><td></td></tr>
<tr><td>sp_for_switch_flag</td><td>2</td><td>u(1)</td></tr>
<tr><td>slice_qs_delta</td><td>2</td><td>se(v)</td></tr>
<tr><td>}</td><td></td><td></td></tr>
<tr><td>if(deblocking_filter_control_present_flag) {</td><td></td><td></td></tr>
<tr><td>disable_deblocking_filter_idc</td><td>2</td><td>ue(v)</td></tr>
<tr><td>if(disable_deblocking_filter_idc != 1) {</td><td></td><td></td></tr>
<tr><td>slice_alpha_c0_offset_div2</td><td>2</td><td>se(v)</td></tr>
<tr><td>slice_beta_offset_div2</td><td>2</td><td>se(v)</td></tr>
<tr><td>}</td><td></td><td></td></tr>
<tr><td>}</td><td></td><td></td></tr>
<tr><td>if(num_slice_groups_minus1 > 0 &&
slice_group_map_type >= 3 && slice_group_map_type <= 5)</td><td></td><td></td></tr>
<tr><td>slice_group_change_cycle</td><td>2</td><td>u(v)</td></tr>
<tr><td>}</td><td></td><td></td></tr>
</table>

Replace Table 7-1 with:

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	Annex I NAL unit type class
0	Unspecified		non-VCL	non-VCL	non-VCL
1	Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp( )	2, 3, 4	VCL	VCL	VCL
2	Coded slice data partition A slice_data_partition_a_layer_rbsp( )	2	VCL	not applicable	not applicable
3	Coded slice data partition B slice_data_partition_b_layer_rbsp( )	3	VCL	not applicable	not applicable
4	Coded slice data partition C slice_data_partition_c_layer_rbsp( )	4	VCL	not applicable	not applicable
5	Coded slice of an IDR picture slice_layer_without_partitioning_rbsp( )	2, 3	VCL	VCL	VCL
6	Supplemental enhancement information (SEI) sei_rbsp( )	5	non-VCL	non-VCL	non-VCL
7	Sequence parameter set seq_parameter_set_rbsp( )	0	non-VCL	non-VCL	non-VCL
8	Picture parameter set pic_parameter_set_rbsp( )	1	non-VCL	non-VCL	non-VCL
9	Access unit delimiter access_unit_delimiter_rbsp( )	6	non-VCL	non-VCL	non-VCL
10	End of sequence end_of_seq_rbsp( )	7	non-VCL	non-VCL	non-VCL
11	End of stream end_of_stream_rbsp( )	8	non-VCL	non-VCL	non-VCL
12	Filler data filler_data_rbsp( )	9	non-VCL	non-VCL	non-VCL
13	Sequence parameter set extension seq_parameter_set_extension_rbsp( )	10	non-VCL	non-VCL	non-VCL
14	Prefix NAL unit prefix_nal_unit_rbsp( )	2	non-VCL	suffix dependent	suffix dependent
15	Subset sequence parameter set subset_seq_parameter_set_rbsp( )	0	non-VCL	non-VCL	non-VCL
16..18	Reserved		non-VCL	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	non-VCL
20	Coded slice extension slice_layer_extension_rbsp( )	2, 3, 4	non-VCL	VCL	VCL
21	Coded slice extension for depth view	2, 3, 4	non-VCL	VCL	VCL

	components /*specified in Annex I */ slice_layer_extension_rbsp( ) /* specified in Annex I */				
22..23	Reserved		non-VCL	non-VCL	VCL
24..31	Unspecified		non-VCL	non-VCL	non-VCL

In 7.4.1, add the following paragraph in the semantics of svc_extension_flag just before the semantics of rbsp_byte[i].

The value of svc_extension_flag shall be equal to 0 for coded video sequences conforming to one or more profiles specified in Annex I. Decoders conforming to one or more profiles specified in Annex I shall ignore (remove from the bitstream and discard) NAL units for which nal_unit_type is equal to 14, 20, or 21 and for which svc_extension_flag is equal to 1.

In 7.4.2.1.3, make the following changes:

Replace the sentence following text in the semantics of chroma_format_idc after "inclusive." with:
When chroma_format_idc is not present and when profile_idc is equal to 138, it shall be inferred to be equal to 0 (4:0:0 chroma format), otherwise, it shall be inferred to be equal to 1 (4:2:0 chroma format).

Substitute each occurrence of "additional_extension2_flag" with "additional_extension3_flag".

Replace the semantics of additional_extension2_data_flag with the following:

additional_extension3_data_flag may have any value. It shall not affect the conformance to profiles specified in Annex A, G, H, or I.

Replace Annex C with:

Annex C\
Hypothetical reference decoder

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

This annex specifies the hypothetical reference decoder (HRD) and its use to check bitstream and decoder conformance.

Two types of bitstreams are subject to HRD conformance checking for this Recommendation | International Standard. The first such type of bitstream, called Type I bitstream, is a NAL unit stream containing only the VCL NAL units and filler data NAL units for all access units in the bitstream. The second type of bitstream, called a Type II bitstream, contains, in addition to the VCL NAL units and filler data NAL units for all access units in the bitstream, at least one of the following:

– additional non-VCL NAL units other than filler data NAL units,

– all leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, and trailing_zero_8bits syntax elements that form a byte stream from the NAL unit stream (as specified in Annex B).

Figure C-1 shows the types of bitstream conformance points checked by the HRD.

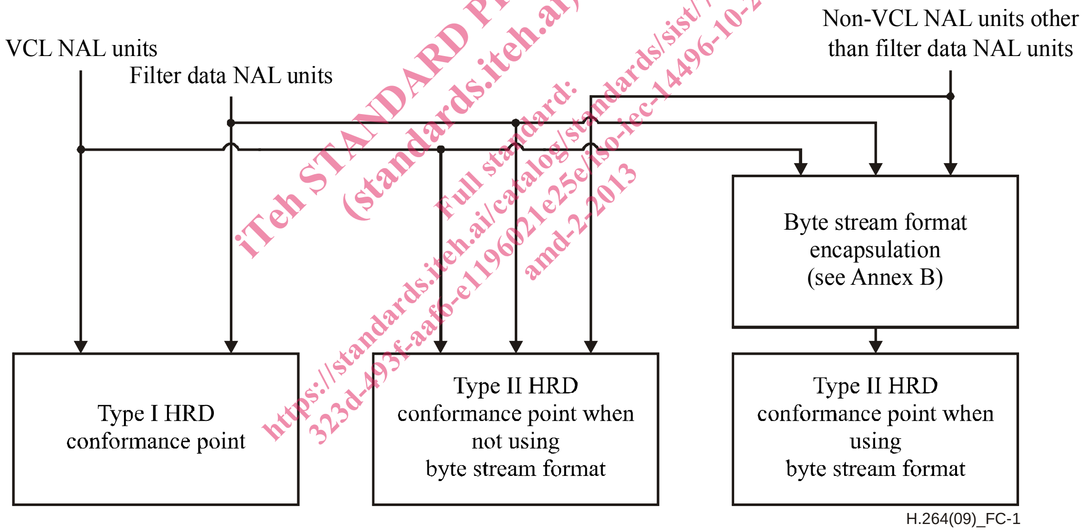

Figure C-1 – Structure of byte streams and NAL unit streams for HRD conformance checks

The syntax elements of non-VCL NAL units (or their default values for some of the syntax elements), required for the HRD, are specified in the semantic subclauses of clause 7, Annexes D and E, and subclauses G.7, G.13, G.14, H.7, H.13, H.14, I.7, I.13, and I.14.

Two types of HRD parameter sets (NAL HRD parameters and VCL HRD parameters) are used. The HRD parameter sets are signalled as follows:

– When the coded video sequence conforms to one or more of the profiles specified in Annex A and the decoding process specified in clauses 2-9 is applied, the HRD parameter sets are signalled through video usability information as specified in subclauses E.1 and E.2, which is part of the sequence parameter set syntax structure.

– When the coded video sequence conforms to one or more of the profiles specified in Annex G and the decoding process specified in Annex G is applied, the HRD parameter sets are signalled through the SVC video usability information extension as specified in subclauses G.14.1 and G.14.2, which is part of the subset sequence parameter set syntax structure.

NOTE 1 – For coded video sequences that conform to both, one or more of the profiles specified in Annex A and one or more of the profiles specified in Annex G, the signalling of the applicable HRD parameter sets is depending on whether the decoding process specified in clauses 2-9 or the decoding process specified in Annex G is applied.

– When the coded video sequence conforms to one or more of the profiles specified in Annex H and the decoding process specified in Annex H is applied, the HRD parameter sets are signalled through the MVC video usability information extension as specified in subclauses H.14.1 and H.14.2, which is part of the subset sequence parameter set syntax structure.

NOTE 2 – For coded video sequences that conform to both, one or more of the profiles specified in Annex A and one or more of the profiles specified in Annex H, the signalling of the applicable HRD parameter sets is depending on whether the decoding process specified in clauses 2-9 or the decoding process specified in Annex H is applied.

– When the coded video sequence conforms to one or more of the profiles specified in Annex I and the decoding process specified in Annex I is applied, the HRD parameter sets are signalled through the MVC video usability information extension as specified in subclauses I.14, which is part of the subset sequence parameter set syntax structure.

NOTE 3 – For coded video sequences that conform to one or more of the profiles specified in Annex A, one or more of the profiles specified in Annex H and one or more of the profiles specified in Annex I, the signalling of the applicable HRD parameter sets is depending on whether the decoding process specified in clauses 2-9, the decoding process specified in Annex H or the decoding process specified in Annex I is applied.

All sequence parameter sets and picture parameter sets referred to in the VCL NAL units, and corresponding buffering period and picture timing SEI messages shall be conveyed to the HRD, in a timely manner, either in the bitstream (by non-VCL NAL units), or by other means not specified in this Recommendation | International Standard.

In Annexes C, D, and E and subclauses G.12, G.13, G.14, H.12, H.13, H.14, I.12, I.13, and I.14, the specification for "presence" of non-VCL NAL units is also satisfied when those NAL units (or just some of them) are conveyed to decoders (or to the HRD) by other means not specified by this Recommendation | International Standard. For the purpose of counting bits, only the appropriate bits that are actually present in the bitstream are counted.

NOTE 3 – As an example, synchronization of a non-VCL NAL unit, conveyed by means other than presence in the bitstream, with the NAL units that are present in the bitstream, can be achieved by indicating two points in the bitstream, between which the non-VCL NAL unit would have been present in the bitstream, had the encoder decided to convey it in the bitstream.

When the content of a non-VCL NAL unit is conveyed for the application by some means other than presence within the bitstream, the representation of the content of the non-VCL NAL unit is not required to use the same syntax specified in this annex.

NOTE 4 – When HRD information is contained within the bitstream, it is possible to verify the conformance of a bitstream to the requirements of this subclause based solely on information contained in the bitstream. When the HRD information is not present in the bitstream, as is the case for all "stand-alone" Type I bitstreams, conformance can only be verified when the HRD data is supplied by some other means not specified in this Recommendation | International Standard.

The HRD contains a coded picture buffer (CPB), an instantaneous decoding process, a decoded picture buffer (DPB), and output cropping as shown in Figure C-2.

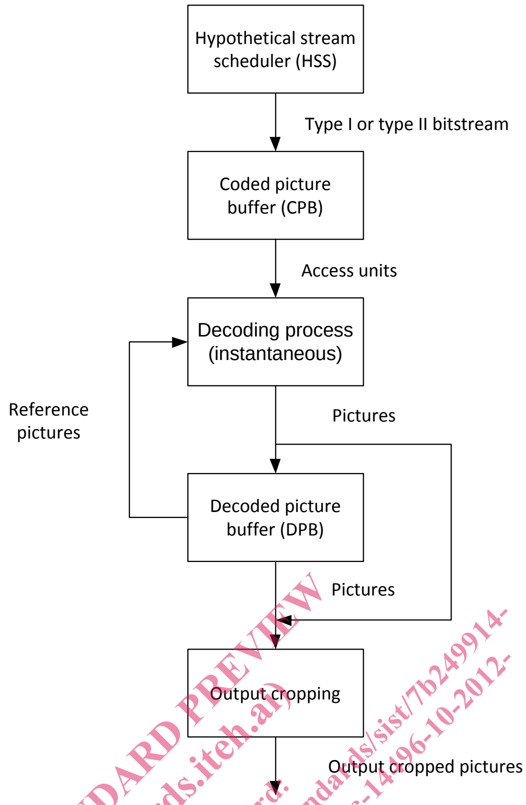

Figure C-2 – HRD buffer model

The CPB size (number of bits) is CpbSize[SchedSelIdx]. The DPB size (number of frame buffers) is Max(1, max_dec_frame_buffering). When the coded video sequence conforms to one or more of the profiles specified in Annex H and the decoding process specified in Annex H is applied, the DPB size is specified in units of view components. When the coded video sequence conforms to one or more of the profiles specified in Annex I and the decoding process specified in Annex I is applied, the DPB size is specified in units of texture view components and depth view components.

The HRD operates as follows. Data associated with access units that flow into the CPB according to a specified arrival schedule are delivered by the HSS. The data associated with each access unit are removed and decoded instantaneously by the instantaneous decoding process at CPB removal times. Each decoded picture is placed in the DPB at its CPB removal time unless it is output at its CPB removal time and is a non-reference picture. When a picture is placed in the DPB it is removed from the DPB at the later of the DPB output time or the time that it is marked as "unused for reference".

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 applied, OutputFlag is set equal to 1.

– Otherwise, if the coded video sequence containing the picture conforms to one or more of the profiles specified in Annex G and the decoding process specified in Annex G is applied, 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 one or more of the profiles specified in Annex H and the decoding process specified in Annex H is applied), the following applies:

