DRAFT AMENDMENT
ISO/IEC 23001-11:2015 DAM 3

ISO/IEC JTC **1**/SC **29** Secretariat: **JISC**

Voting begins on:
2017-10-24

Voting terminates on:
2018-01-16

Information technology — MPEG systems technologies —

Part 11:
Energy-efficient media consumption (green metadata)

AMENDMENT 3

Technologies de l'information — Technologies des systèmes MPEG —

Partie 11: Consommation des supports éconergétiques (métadonnées vertes)

AMENDEMENT 3

ICS: 35.040.40

Reference number
ISO/IEC 23001-11:2015/DAM 3:2017(E)

INTERNATIONAL STANDARD

Information technology – MPEG Systems Technologies – Part 11: Energy-Efficient Media Consumption (Green Metadata)

Amendment 3

AVC and HEVC metrics for complexity prediction models

1) New syntax and semantics in §5.2 Complexity metrics for decoder-power reduction

In §5.2.1, replace:

“With respect to the functional architecture in Figure 1, the green-metadata generator provides CMs that indicate the picture-decoding complexity of an AVC bitstream to the decoder.”

with:

“With respect to the functional architecture in Figure 1, the green-metadata generator provides CMs that indicate the picture-decoding complexity of an AVC or HEVC bitstream to the decoder.”

In §5.2.2, replace:

“The syntax for the CMs is as follows:”

with

“The syntax for the AVC CMs is as follows:”

In §5.2.2, replace the existing table with:

	Size (bits)	**Descriptor**
period_type	8	unsigned integer
if (period_type == 2) || (period_type == 7) {		
num_seconds	16	unsigned integer
}		
else if (period_type == 3) || (period_type == 8) {		
num_pictures	16	unsigned integer
}		
if (period_type == 8) {		
temporal_map		
for (t=0; t<8; t++) {		
if ((temporal_map>>t)%2 == 1)		
num_pictures_in_temporal_layers[t]		
}		
}		
if (period_type <= 3) {		
percent_non_zero_8x8_blocks	8	unsigned integer
percent_intra_predicted_macroblocks	8	unsigned integer

percent_six_tap_filterings	8	unsigned integer
percent_alpha_point_deblocking_instances	8	unsigned integer
}		
else if (period_type == 4) {		
for (i=0; i<= num_slice_groups_minus1; i++) {		
num_slices_minus1[i]	16	unsigned integer
}		
for (i=0; i<= num_slice_groups_minus1; i++) {		
for (j=0; j<=num_slices_minus1[i]; j++) {		
first_mb_in_slice[i][j]	16	unsigned integer
percent_non_zero_8x8_blocks[i][j]	8	unsigned integer
percent_intra_predicted_macroblocks[i][j]	8	unsigned integer
percent_six_tap_filterings[i][j]	8	unsigned integer
percent_alpha_point_deblocking_instances[i][j]	8	unsigned integer
}		
}		
}		
else if (period_type >= 5) && (period_type <= 8) {		
num_layers_minus1	16	unsigned integer
for (l=0; l<= num_layers_minus1; l++) {		
picture_parameter_set_id[l]	8	unsigned integer
priority_id[l]	6	unsigned integer
dependency_id[l]	3	unsigned integer
quality_id[l]	4	unsigned integer
temporal_id[l]	3	unsigned integer
percent_non_zero_8x8_blocks[l]	8	unsigned integer
percent_intra_predicted_macroblocks[l]	8	unsigned integer
percent_six_tap_filterings[l]	8	unsigned integer
percent_alpha_point_deblocking_instances[l]	8	unsigned integer
}		
}		

In §5.2.2, add:

"The syntax for the HEVC CMs is as follows:"

and add the table:

	Size	**Descriptor**
period_type	8	unsigned integer
if (period_type == 2) {		
num_seconds	16	unsigned integer
}		
else if (period_type == 3) {		
num_pictures	16	unsigned integer
}		
if (period_type <= 3) {		
percent_non_zero_blocks_surface	8	unsigned integer

if (percent_non_zero_blocks_surface != 0) {		
percent_8x8_blocks_in_non_zero_surface	8	unsigned integer
percent_16x16_blocks_in_non_zero_surface	8	unsigned integer
percent_32x32_blocks_in_non_zero_surface	8	unsigned integer
}		
percent_intra_predicted_blocks_surface	8	unsigned integer
if (percent_intra_predicted_blocks_surface == 255) {		
percent_planar_blocks_in_intra_surface	8	unsigned integer
percent_DC_blocks_in_intra_surface	8	unsigned integer
percent_angularHorV_blocks_in_intra_surface	8	unsigned integer
}		
else {		
percent_blocks_a_c_filterings	8	unsigned integer
percent_blocks_h_b_filterings	8	unsigned integer
percent_blocks_f_i_k_q_filterings	8	unsigned integer
percent_blocks_j_filterings	8	unsigned integer
percent_blocks_e_g_p_r_filterings	8	unsigned integer
percent_blocks_d_n_filterings	8	unsigned integer
}		
percent_weak_deblocking_instances	8	unsigned integer
}		
else if (period_type == 4) {		
max_num_slices_tiles_minus1	16	unsigned integer
for (t=0; t<=max_num_slices_tiles_minus1; t++) {		
first_ctb_in_slice_or_tile[t]	16	unsigned integer
percent_non_zero_blocks_surface[t]	8	unsigned integer
if (percent_non_zero_blocks_surface[t] != 0) {		
percent_8x8_blocks_in_non_zero_surface[t]	8	unsigned integer
percent_16x16_blocks_in_non_zero_surface[t]	8	unsigned integer
percent_32x32_blocks_in_non_zero_surface[t]	8	unsigned integer
}		
percent_intra_predicted_blocks_surface[t]	8	unsigned integer
if (percent_intra_predicted_blocks_surface[t] == 255) {		
percent_planar_blocks_in_intra_surface[t]	8	unsigned integer
percent_DC_blocks_in_intra_surface[t]	8	unsigned integer
percent_angularHorV_blocks_in_intra_surface[t]	8	unsigned integer
}		
else {		
percent_blocks_a_c_filterings[t]	8	unsigned integer
percent_blocks_h_b_filterings[t]	8	unsigned integer
percent_blocks_f_i_k_q_filterings[t]	8	unsigned integer
percent_blocks_j_filterings[t]	8	unsigned integer
percent_blocks_e_g_p_r_filterings[t]	8	unsigned integer
percent_blocks_d_n_filterings[t]	8	unsigned integer
}		
percent_weak_deblocking_instances[t]	8	unsigned integer

}		
}		

In §5.2.3, replace:

"SEI messages can be used to signal Green Metadata in an AVC stream. The Green Metadata SEI message payload type is specified in ISO/IEC 14496-10:2014/Amd. 2. The complete syntax of the Green Metadata SEI message payload is specified in Annex A."

with

"SEI messages can be used to signal Green Metadata in an AVC or HEVC stream. The Green Metadata SEI message payload types are specified in ISO/IEC 14496-10 and ISO/IEC 23008-2. The complete syntax of the Green Metadata SEI message payloads is specified in Annex A of this document."

In §5.2.4, replace the existing text with the following text:

5.2.4.1 AVC semantics

The semantics of various terms are defined below.

period_type specifies the type of upcoming period over which the four complexity metrics are applicable and is defined in the following table.

Value	Description
0x00	complexity metrics are applicable to a single picture
0x01	complexity metrics are applicable to all pictures in decoding order, up to (but not including) the picture containing the next I slice
0x02	complexity metrics are applicable over a specified time interval in seconds
0x03	complexity metrics are applicable over a specified number of pictures counted in decoding order
0x04	complexity metrics are applicable to a single picture with slice granularity
0x05	complexity metrics are applicable to a single picture with scalable layer granularity
0x06	complexity metrics are applicable to all pictures in decoding order, up to (but not including) the picture containing the next I slice in the base layer with scalable layer granularity
0x07	complexity metrics are applicable over a specified time interval in seconds with scalable layer granularity
0x08	complexity metrics are applicable over a specified number of pictures counted in decoding order with scalable layer granularity
0x09-0xFF	reserved

num_seconds indicates the number of seconds over which the complexity metrics are applicable when period_type is 2 or 7.

num_pictures specifies the number of pictures, counted in decoding order, over which the complexity metrics are applicable when period_type is 3 or 8. When period_type is 8, this is a default number of pictures for each temporal layer, which can be overidden using temporal_map flags.

num_pics_in_period specifies the number of pictures in the specified period. When period_type is 0 or 4, then num_pics_in_period is 1. When period_type is 1, then num_pics_in_period is determined by counting the pictures in decoding order up to (but not including) the one containing the next I slice.

When period_type is 2, then num_pics_in_period is determined from the frame rate. When period_type is 3, then num_pics_in_period is equal to num_pictures.

temporal_map specifies which temporal layer has a different number of pictures from num_pictures in the specified period, when period_type is 8.

num_pictures_in_temporal_layer[t] specifies the number of pictures in the specified period for the t^{th} temporal layer when period_type is 8. When not present, it is equal to num_pictures.

num_pics_in_period_for_temporal_layer[t] specifies the number of pictures in the specified period for the t^{th} temporal layer. When period_type is 5 then num_pics_in_period_for_temporal_layer[t] is 1. When period_type is 6, then num_pics_in_period_for_temporal_layer[t] is determined by counting the pictures associated to the t^{th} temporal layer in decoding order up to (but not including) the one containing the next I slice. When period_type is 7, then num_pics_in_period_for_temporal_layer[t] is determined from the frame rate associated to the t^{th} temporal layer. When period_type is 8, then num_pics_in_period_for_temporal_layer[t] is equal to num_pictures_in_temporal_layer[t]

total_num_macroblocks_pic(n) set to the value of the AVC variable PicSizeInMbs for the n^{th} picture within the specified period, where 1 <= n<= num_pics_in_period.

total_num_macroblocks_in_period indicates the total number of macroblocks that are coded in the specified period. Determined by the following computation:

$$\sum_{n=1}^{\text{num\_pics\_in\_period}} \text{total\_num\_macroblocks\_pic}(n) \qquad (5\text{-}1)$$

num_non_zero_8x8_blocks indicates the number of non-zero 8x8 blocks in the specified period.

percent_non_zero_8x8_blocks indicates the percentage of non-zero 8x8 blocks in the specified period and is defined as follows:

$$\text{percent\_non\_zero\_8x8\_blocks} = \text{Floor}\left(\frac{\text{num\_non\_zero\_8x8\_blocks}}{\text{total\_num\_macroblocks\_in\_period} * 4} * 255\right) \qquad (5\text{-}2)$$

num_intra_predicted_macroblocks indicates the number of macroblocks using Intra prediction modes in the specified period.

percent_intra_predicted_macroblocks indicates the percentage of macroblocks using Intra prediction modes in the specified period and is defined as follows:

$$\text{percent\_intra\_predicted\_macroblocks} = \text{Floor}\left(\frac{\text{num\_intra\_predicted\_macroblocks}}{\text{total\_num\_macroblocks\_in\_period}} * 255\right) \qquad (5\text{-}3)$$

num_six_tap_filterings indicates the number of 6-tap filterings (STFs), as defined in ISO/IEC 14496-10, within the specified period.

max_num_six_tap_filterings_in_period indicates the maximum number of STFs that could occur within the specified period and is defined as follows:

max_num_six_tap_filterings_in_period = (1664 * total_num_macroblocks_in_layer_in_period) (5-4)

percent_six_tap_filterings indicates the percentage of STFs in the specified period and is defined as follows:

$$\text{percent\_six\_tap\_filterings} = \text{Floor}\left(\frac{\text{num\_six\_tap\_filterings}}{\text{max\_num\_six\_tap\_filterings\_in\_period}} * 255\right) \qquad (5\text{-}5)$$

num_alpha_point_deblocking_instances indicates the number of Alpha-Point Deblocking Instances (APDIs) in the specified period. Using the notation in ISO/IEC 14496-10, this is equivalent to the total number of filtering operations applied to produce filtered samples of the type p'_0 or q'_0, in the specified period.

max_num_alpha_point_deblocking_instances_in_period indicates the maximum number of APDIs that could occur within the specified period. Set as follows:

$$\text{max\_num\_alpha\_point\_deblocking\_instances\_in\_period} = 128 * \text{chroma\_format\_multiplier} * \text{total\_num\_macroblocks\_in\_layer\_in\_period} \quad (5\text{-}6)$$

where chroma_format_multiplier depends on the AVC variables separate_colour_plane_flag and chroma_format_idc as shown in the following table.

chroma_format_multiplier	separate_colour_plane_flag	chroma_format_idc	Comment
1	0	0	monochrome
1.5	0	1	4:2:0 sampling
2	0	2	4:2:2 sampling
3	0	3	4:4:4 sampling
3	1	any value	separate colour plane

percent_alpha_point_deblocking_instances indicates the percentage of APDIs in the specified period and is defined as follows:

$$\text{percent\_alpha\_point\_deblocking\_instances} = \text{Floor}\left(\frac{\text{num\_alpha\_point\_deblocking\_instances}}{\text{max\_num\_alpha\_point\_deblocking\_instances\_in\_period}} * 255\right) \quad (5\text{-}7)$$

num_slices_minus1 plus 1 specifies the number of slices per slice_group in the picture.

first_mb_in_slice[i][j] specifies the first macroblock number in the slice[i][j].

total_num_macroblocks_in_slice[i][j] indicates the total number of macroblocks that are coded in the slice[i][j]. Determined by the following computation:

– If num_slice_groups_minus1 is equal to 0:

```
if (j<num_slices_minus1[0])
  total_num_macroblocks_in_slice[0][j]=first_mb_in_slice[0][j+1] - first_mb_in_slice[0][j]
else
  total_num_macroblocks_in_slice[0][j]=PicSizeInMbs - first_mb_in_slice[0][j]
```

- Otherwise (num_slice_groups_minus1 is not equal to 0), and after derivation of the macroblock to slice group map (MbToSliceGroupMap) as specified in subclause “Specification for conversion of map unit to slice group map to macroblock to slice group map” in ISO/IEC 14496-10,

```
if (j<num_slices_minus1[i])
   k=0;
    for ( n=first_mb_in_slice[i][j]; n< first_mb_in_slice[i][j+1]; n++ )
          if ( MbToSliceGroupMap[first_mb_in_slice[i][j]] == MbToSliceGroupMap[n] )
                 k++;
   total_num_macroblocks_in_slice[i][j]=k;
else
    k=0;
```

```
for ( n=first_mb_in_slice[i][j]; n< picSizeInMbs; n++ )
        if ( MbToSliceGroupMap[first_mb_in_slice[i][j]] == MbToSliceGroupMap[n] )
                k++;
total_num_macroblocks_in_slice[i][j]=k;                                    (5-8)
```

num_non_zero_8x8_blocks[i][j] indicates the number of non-zero 8x8 blocks in the slice[i][j].

percent_non_zero_8x8_blocks[i][j] indicates the percentage of non-zero 8x8 blocks in the slice[i][j] and is defined as follows:.

$$\text{percent\_non\_zero\_8x8\_blocks[i][j]} = \text{Floor}\left(\frac{\text{num\_non\_zero\_8x8\_blocks[i][j]}}{\text{total\_num\_macroblocks\_in\_slice[i][j]}*4} * 255\right) \quad (5\text{-}9)$$

num_intra_predicted_macroblocks[i][j] indicates the number of macroblocks using Intra prediction modes in the slice[i][j].

percent_intra_predicted_macroblocks[i][j] indicates the percentage of macroblocks using Intra prediction modes in the slice[i][j] and is defined as follows:

$$\text{percent\_intra\_coded\_macroblocks[i][j]} = \text{Floor}\left(\frac{\text{num\_intra\_predicted\_macroblocks[i][j]}}{\text{total\_num\_macroblocks\_in\_slice[i][j]}} * 255\right) \quad (5\text{-}10)$$

num_six_tap_filterings[i][j] indicates the number of 6-tap filterings (STFs), as defined in ISO/IEC 14496-10, within the slice[i][j].

max_num_six_tap_filterings_in_slice[i][j] indicates the maximum number of STFs that could occur in the slice[i][j]. Set to the value (1664 * total_num_macroblocks_in_slice[i][j]).

percent_six_tap_filterings[i][j] indicates the percentage of STFs in the specified slice[i][j] and is defined as follows:

$$\text{percent\_six\_tap\_filterings[i][j]} = \text{Floor}\left(\frac{\text{num\_six\_tap\_filterings[i][j]}}{\text{max\_num\_six\_tap\_filterings\_in\_slice[i][j]}} * 255\right) \quad (5\text{-}11)$$

num_alpha_point_deblocking_instances indicates the number of Alpha-Point Deblocking Instances (APDIs) in slice[i][j].

max_num_alpha_point_deblocking_instances_in_slice[i][j] indicates the maximum number of APDIs that could occur in the slice[i][j]. Set as follows:

max_num_alpha_point_deblocking_instances_in_slice[i][j] =
128 * chroma_format_multiplier * total_num_macroblocks_in_slice[i][j]

(5-12)

percent_alpha_point_deblocking_instances[i][j] indicates the percentage of APDIs in the specified slice[i][j] and is defined as follows:

$$\text{percent\_alpha\_point\_deblocking\_instances[i][j]} = \text{Floor}\left(\frac{\text{num\_alpha\_point\_deblocking\_instances[i][j]}}{\text{max\_num\_alpha\_point\_deblocking\_instances\_in\_slice[i][j]}} * 255\right)$$

(5-13)

num_layers_minus1 plus 1 specifies the number of scalable layers in the associated picture or in the specified period.

pic_parameter_set_id[l] specifies the picture parameter set in use for the l[th] scalable layer. The value of pic_parameter_set_id[l] shall be in the range of 0 to 255, inclusive (as specified in subclause "Slice header semantics" in ISO/IEC 14496-10).

priority_id[l] - specifies a priority identifier for the NAL unit in the l[th] scalable layer. The value of priority_id[l] shall be in the range of 0 to 63, inclusive (as specified in subclause "Slice data in scalable extension semantics" in ISO/IEC 14496-10).

dependency_id[l] specifies a dependency identifier for the NAL unit in the l[th] scalable layer. The value of dependency_id[l] shall be in the range of 0 to 7, inclusive (as specified in subclause "Slice data in scalable extension semantics" in ISO/IEC 14496-10).

quality_id[l] specifies a quality identifier for the NAL unit in the l[th] scalable layer. The value of quality_id[l] shall be in the range of 0 to 15, inclusive (as specified in subclause "Slice data in scalable extension semantics" in ISO/IEC 14496-10).

temporal_id[l] specifies a temporal identifier for the NAL unit in the l[th] scalable layer. The value of temporal_id[l] shall be in the range of 0 to 7, inclusive (as specified in subclause "Slice data in scalable extension semantics" in ISO/IEC 14496-10).

total_num_macroblocks_in_layer[l] indicates the total number of macroblocks in the lth scalable layer and determined after derivation of the number of macroblock associated with the pic_parameter_set_id[l], as specified in subclause "Slice header semantics" in ISO/IEC 14496-10.

total_num_macroblocks_in_layer_in_period[l] indicates the total number of macroblocks in the lth scalable layer in the specified period and is defined as follows

$$\text{total\_num\_macroblocks\_in\_layer\_in\_period}[l] = \text{total\_num\_macroblocks\_in\_layer}[l] * \text{num\_pics\_in\_period\_for\_temporal\_layers}[\text{temporal\_id}[l]] \quad (5\text{-}14)$$

num_non_zero_8x8_blocks[l] indicates the number of non-zero 8x8 blocks in the l[th] scalable layer in the specified period.

percent_non_zero_8x8_blocks[l] indicates the percentage of non-zero 8x8 blocks in the l[th] scalable layer and is defined as follows:

$$\text{percent\_non\_zero\_8x8\_blocks}[l] = \text{Floor}\left(\frac{\text{num\_non\_zero\_8x8\_blocks}[l]}{\text{total\_num\_macroblocks\_in\_layer\_in\_period}[l]*4} * 255\right) \quad (5\text{-}15)$$

num_intra_predicted_macroblocks[l] indicates the number of macroblocks using Intra prediction modes in the l[th] scalable layer in the specified period.

percent_intra_predicted_macroblocks[l] indicates the percentage ofmacroblocks using Intra prediction modes in the l[th] scalable layer and is defined as follows:

$$\text{percent\_intra\_predicted\_macroblocks}[l] = \text{Floor}\left(\frac{\text{num\_intra\_predicted\_macroblocks}[l]}{\text{total\_num\_macroblocks\_in\_layer\_in\_period}[l]} * 255\right) \quad (5\text{-}16)$$

num_six_tap_filterings[l] indicates the number of 6-tap filterings (STFs), as defined in ISO/IEC 14496-10, within the l[th] scalable layer in the specified period.

max_num_six_tap_filterings_in_layer_in_period[l] indicates the maximum number of STFs that could occur in the l[th] scalable layer in the specified period. Set to the value (1664 * total_num_macroblocks_in_layer_in_period [l]).

percent_six_tap_filterings[l] indicates the percentage of STFs in the specified l[th] scalable layer in the specified period and is defined as follows:

$$\text{percent\_six\_tap\_filterings}[l] = \text{Floor}\left(\frac{\text{num\_six\_tap\_filterings}[l]}{\text{max\_num\_six\_tap\_filterings\_in\_layer\_in\_period}[l]} * 255\right) \quad (5\text{-}17)$$

num_alpha_point_deblocking_instances[l] indicates the number of Alpha-Point Deblocking Instances (APDIs) in the i[th] scalable layer in the specified period.

max_num_alpha_point_deblocking_instances_in_layer_in_period[l] indicates the maximum number of APDIs that could occur in the l[th] scalable layer in the specified period. Set as follows:

max_num_alpha_point_deblocking_instances_in_layer[l] =
128 * chroma_format_multiplier * total_num_macroblocks_in_layer_in_period [l]

(5-18)

percent_alpha_point_deblocking_instances[l] indicates the percentage of APDIs in the specified l^{th} scalable layer in the specified period and is defined as follows:

$$\text{percent\_alpha\_point\_deblocking\_instances[l]} = \text{Floor}\left(\frac{\text{num\_alpha\_point\_deblocking\_instances[l]}}{\text{max\_num\_alpha\_point\_deblocking\_instances\_in\_layer\_in\_period[l]}} * 255\right)$$

(5-19)

5.2.4.2 HEVC semantics

The semantics of various terms are defined below.

period_type specifies the type of upcoming period over which the four complexity metrics are applicable and is defined in the following table.

Value	**Description**
0x00	complexity metrics are applicable to a single picture
0x01	complexity metrics are applicable to all pictures in decoding order, up to (but not including) the picture containing the next I slice
0x02	complexity metrics are applicable over a specified time interval in seconds
0x03	complexity metrics are applicable over a specified number of pictures counted in decoding order
0x04	complexity metrics are applicable to a single picture with slice or tile granularity
0x05-0xFF	reserved

num_seconds when period_type is 2, num_seconds indicates the number of seconds over which the complexity metrics are applicable.

num_pictures when period_type is 3, num_pictures specifies the number of pictures, counted in decoding order, over which the complexity metrics are applicable.

num_pics_in_period specifies the number of pictures in the specified period. When period_type is 0, then num_pics_in_period is 1. When period_type is 1, then num_pics_in_period is determined by counting the pictures in decoding order up to (but not including) the one containing the next I slice. When period_type is 2, then num_pics_in_period is determined from the frame rate. When period_type is 3, then num_pics_in_period is equal to num_pictures.

total_num_4x4_blocks_pic(n) set to the value of the HEVC variable

$$\text{PicSizeInCtbsY} * (1 << (\text{CtbLog2SizeY}-2))^2 \quad (5\text{-}20)$$

for the n^{th} picture within the specified period, where 1 <= n <= num_pics_in_period.

total_num_4x4_blocks_in_period indicates the total number of 4x4 blocks that are coded in the specified period.

Determined by the following computation:

$$\sum_{n=0}^{num\_pics\_in\_period} \text{total\_num\_4x4\_blocks\_pic(n)}$$

(5-21)

num_non_zero_4x4_blocks indicates the number of non-zero 4x4 blocks in the specified period.