INTERNATIONAL STANDARD ISO/IEC 14496-2:1999
TECHNICAL CORRIGENDUM 2

Published 2001-02-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Coding of audio-visual objects —

Part 2:
Visual

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Codage des objets audiovisuels —

Partie 2: Codage visuel

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to International Standard ISO/IEC 14496-2:1999 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ICS 35.040 **Ref. No. ISO/IEC 14496-2:1999/Cor.2:2001(E)**

Printed in Switzerland

In clause 3 Definitions, add the following definition with appropriate numbering in alphabetical order

"

3.XXX reference layer: A layer to be referenced for prediction in a scalable hierarchy. The video_object_id of the reference layer should be the same value as the video_object_id of the enhancement layer. The ref_layer_id of the enhancement layer is set to the same value as the video_object_layer_id of the reference layer.

"

Replace subclause 6.1.3.5 I-VOPs and group of VOPs with

"

6.1.3.5 I-VOPs and group of VOPs

I-VOPs are intended to assist random access into the sequence. Applications requiring random access, fast-forward playback, or fast reverse playback may use I-VOPs relatively frequently.

I-VOPs may also be used at scene cuts or other cases where motion compensation is ineffective.

Group of VOP (GOV) header is an optional header that can be used immediately before a coded I-VOP to indicate to the decoder:

1) the modulo part (i.e. the full second units) of the time base for the next VOP to be displayed after having decoded a GOV header
2) if the first consecutive B-VOPs immediately following the coded I-VOP can be reconstructed properly in the case of a random access.

In a non scalable bitstream or the base layer of a scalable bitstream, the first coded VOP following a GOV header shall be a coded I-VOP.

"

In subclause 6.2.1 Start code, replace

"

1. Configuration information

   a. Global configuration information, referring to the whole group of visual objects that will be simultaneously decoded and composited by a decoder (VisualObjectSequence()).

   b. Object configuration information, referring to a single visual object (VO). This is associated with VisualObject().

   c. Object layer configuration information, referring to a single layer of a single visual object (VOL) VisualObjectLayer()

"

with

"

1. Configuration information

   a. Global configuration information, referring to the whole group of visual objects that will be simultaneously decoded and composited by a decoder (VisualObjectSequence()).

   b. Object configuration information, referring to a single visual object (VO). This is associated with VisualObject().

   c. Object layer configuration information, referring to a single layer of a single visual object (VOL) VideoObjectLayer().

"

In subclause 6.2.1 Start codes, replace the following row of Table 6-3

"

Reserved	C3 – C5

"

with

"

Stuffing_start_code	C3
Reserved	C4 – C5

"

In subclause 6.2.3 VideoObjectLayer, replace the following rows

"

…		
scalability	1	bslbf
if (scalability) {		
hierarchy_type	1	bslbf
ref_layer_id	4	uimsbf
ref_layer_sampling_direc	1	bslbf
hor_sampling_factor_n	5	uimsbf
hor_sampling_factor_m	5	uimsbf
vert_sampling_factor_n	5	uimsbf
vert_sampling_factor_m	5	uimsbf
enhancement_type	1	bslbf
if(video_object_layer == "binary" && hierarchy_type== '0') {		
use_ref_shape	1	bslbf
use_ref_texture	1	bslbf
shape_hor_sampling_factor_n	5	uimsbf
shape_hor_sampling_factor_m	5	uimsbf
shape_vert_sampling_factor_n	5	uimsbf
shape_vert_sampling_factor_m	5	uimsbf
}		
}		
}		
else {		
if(video_object_layer_verid !="0001") {		
scalability	1	bslbf
if(scalability) {		
shape_hor_sampling_factor_n	5	uimsbf
shape_hor_sampling_factor_m	5	uimsbf
shape_vert_sampling_factor_n	5	uimsbf
shape_vert_sampling_factor_m	5	uimsbf
}		
}		
resync_marker_disable	1	bslbf
}		

"

with

"

…		
scalability	1	bslbf
if (scalability) {		
hierarchy_type	1	bslbf
ref_layer_id	4	uimsbf
ref_layer_sampling_direc	1	bslbf
hor_sampling_factor_n	5	uimsbf
hor_sampling_factor_m	5	uimsbf
vert_sampling_factor_n	5	uimsbf
vert_sampling_factor_m	5	uimsbf
enhancement_type	1	bslbf
if(video_object_layer == “binary” && hierarchy_type== ‘0’) {		
use_ref_shape	1	bslbf
use_ref_texture	1	bslbf
shape_hor_sampling_factor_n	5	uimsbf
shape_hor_sampling_factor_m	5	uimsbf
shape_vert_sampling_factor_n	5	uimsbf
shape_vert_sampling_factor_m	5	uimsbf
}		
}		
}		
else {		
if(video_object_layer_verid !=”0001”) {		
scalability	1	bslbf
if(scalability) {		
ref_layer_id	4	uimsbf
shape_hor_sampling_factor_n	5	uimsbf
shape_hor_sampling_factor_m	5	uimsbf
shape_vert_sampling_factor_n	5	uimsbf
shape_vert_sampling_factor_m	5	uimsbf
}		
}		
resync_marker_disable	1	bslbf
}		

"

In subclause 6.2.3 Video Object Layer, replace the following rows

"

do {		
if (next_bits() == group_of_vop_start_code)		
Group_of_VideoObjectPlane()		
VideoObjectPlane()		
} while ((next_bits() == group_of_vop_start_code) || (next_bits() == vop_start_code))		
} else {		
short_video_header = 1		
do {		
video_plane_with_short_header()		
} while(next_bits() == short_video_start_marker)		
}		
}		

"

with

"

do {		
if (next_bits() == group_of_vop_start_code)		
Group_of_VideoObjectPlane()		
VideoObjectPlane()		
if ((preceding_vop_coding_type == "B" || preceding_vop_coding_type == "S" || video_object_layer_shape != "rectangular") && next_bits() == stuffing_start_code) {		
stuffing_start_code	32	bslbf
while (next_bits() != '0000 0000 0000 0000 0000 0001')		
stuffing_byte	8	bslbf
}		
} while ((next_bits() == group_of_vop_start_code) || (next_bits() == vop_start_code))		
} else {		
short_video_header = 1		
do {		
video_plane_with_short_header()		
} while(next_bits() == short_video_start_marker)		
}		
}		
NOTE — preceding_vop_coding_type has the same value as vop_coding_type in the immediately preceding VideoObjectPlane() in the decoding order.		

"

In subclause 6.2.5.1 Complexity Estimation Header, replace the last 8 rows of the read_vop_complexity_estimation_header() syntax

"

if (npm)	**dcecs_npm**	8	uimsbf
if (forw_back_mc_q)	**dcecs_forw_back_q**	8	uimsbf
if (halfpel2)	**dcecs_halfpel2**	8	uimsbf
if (halfpel4)	**dcecs_halfpel4**	8	uimsbf
if (interpolate_mc_q)	**dcecs_interpolate_mc_q**	8	uimsbf
}			
}			
}			

"

with

"

if (npm)	**dcecs_npm**	8	uimsbf
if (forw_back_mc_q)	**dcecs_forw_back_mc_q**	8	uimsbf
if (halfpel2)	**dcecs_halfpel2**	8	uimsbf
if (halfpel4)	**dcecs_halfpel4**	8	uimsbf
if (interpolate_mc_q)	**dcecs_interpolate_mc_q**	8	uimsbf
}			
}			
}			

"

In subclause 6.3.3, replace the semantics of use_ref_shape with

"

use_ref_shape: This is one bit flag which indicate procedure to decode binary shape for spatial scalability. If it is set to '0', scalable shape coding should be used. If it is set to '1' and enhancement_type is set to '0', no shape data is decoded and up-sampled binary shape of reference_layer should be used for enhancement layer. If enhancement_type is set to '1' and this flag is set to '1',binary shape of enhancement layer should be decoded as the same non-scalable decoding process. When video_object_layer_verid == '0001', the value of use_ref_shape is set to '1'.

"

In subclause 6.3.3, replace the semantics of use_ref_texture with

"

use_ref_texture: Reserved flag for future extension. This flag shall be 0 in the case of video_object_layer_ver_id is "0001" or "0010".

"

In subclause 6.3.3 Video Object Layer, add the following paragraphs at the end of the subclause

"

stuffing_start_code: This is the bit string '000001C3' in hexadecimal. It is used in conjunction with possibly following stuffing_byte(s) for the purpose of stuffing bits to guaranty the VBV buffer regulation.

stuffing_byte: This is the 8-bit string in which the value is '11111111'.

"

In subclause 6.3.5, replace the semantics of modulo_time_base with

"

modulo_time_base: This value represents the local time base in one second resolution units (1000 milliseconds). It consists of a number of consecutive '1' followed by a '0'. Each '1' represents a duration of one second that have elapsed. For I-, S(GMC)-, and P-VOPs of a non scalable bitstream and the base layer of a scalable bitstream, the number of '1's indicate the number of seconds elapsed since the synchronization point marked by time_code of the previous GOV header or by modulo_time_base of the previously decoded I-, S(GMC)-, or P-VOP, in decoding order. For B-VOP of a non scalable bitstream and a base layer of a scalable bitstream, the number of '1's indicates the number of seconds elapsed since the synchronization point marked in the previous GOV header, or I-VOP, S(GMC)-VOP, or P-VOP, in display order. For I-, P-, or B-VOPs of enhancement layer of scalable bitstream, the number of '1's indicate the number of seconds elapsed since the synchronization point marked in the previous GOV header, I-VOP, P-VOP, or B-VOP, in display order.

"

In subclause 6.3.5, replace the semantics of vop_vertical_mc_spatial_ref with

"

vop_vertical_mc_spatial_ref: This is a 13-bit signed integer which specifies, in pixel units, the vertical position of the top left of the rectangle defined by vertical size of vop_height. The value of vop_vertical_mc_spatial_ref shall be divisible by two for progressive and divisible by four for interlaced motion compensation. This is used for decoding and for picture composition.

"

In subclause 6.3.5, replace the semantics of resync_marker with

"

resync_marker: This is a binary string of at least 16 zero's followed by a one'0 0000 0000 0000 0001'. For an I-VOP or a VOP where video_object_layer_shape has the value "binary_only", the resync marker is 16 zeros followed by a one. The length of this resync marker is dependent on the value of vop_fcode_forward, for a P-VOP or a S(GMC)-VOP, and the larger value of either vop_fcode_forward and vop_fcode_backward for a B-VOP. For a P-VOP and a S(GMC)-VOP, the resync_marker is (15+fcode) zeros followed by a one; for a B-VOP, the resync_marker is max(15+fcode,17) zeros followed by a one. It is only present when resync_marker_disable flag is set to '0'. A resync marker shall only be located immediately before a macroblock and aligned with a byte

"

In subclause 6.3.5 Video Object Plane and Video Plane with Short Header, replace the semantics of header_extension_code with

"

header_extension_code: This is a 1-bit flag which when set to '1' indicates the presence of additional fields in the header. When header_extension_code is set to '1', modulo_time_base, vop_time_increment and vop_coding_type are also included in the video packet header. If video_object_layer_shape is not "rectangular", VOP header fields used for the shape decoding (vop_width, vop_height, vop_horizontal_mc_spatial_ref, vop_vertical_mc_spatial_ref, change_conv_ratio_disable and vop_shape_coding_type) are also present. if video_object_layer_shape is not "binary only", intra_dc_vlc_thr is also present**.** Furthermore, if the vop_coding_type is equal to either a P, S or B VOP, the appropriate fcodes are also present. Additionally, if the current VOP is an S(GMC)-VOP, sprite_trajectory() is included. And if reduced_resolution_vop_enable is equal to one, vop_reduced_resolution is also present.

"

In subclause 6.3.6 Macroblock related, replace the semantics of cbpb with

"

cbpb: This is a 3 to 6 bit code representing coded block pattern in B-VOPs, if indicated by modb. Each bit in the code represents a coded/no coded status of a block; the leftmost bit corresponds to the top left block in the macroblock. For each non-transparent blocks with coefficients, the corresponding bit in the code is set to '1'. In case no coefficients are coded for all the non-transparent blocks in the macroblock, modb shall be set to the value indicating cbpb is not present (i.e. modb=='1' or '01') and cbpb shall not be included in the bitstream for this macroblock.

"

In subclause 7.4.1.2 Other coefficients, replace

"

When short_video_header is 0, the variable length code table is different for intra blocks and inter blocks. The most commonly occurring EVENTs for the luminance and chrominance components of intra blocks in this case are decoded by referring to Table B-16. The most commonly occurring EVENTs for the luminance and chrominance components of inter blocks in this case are decoded by referring to Table B-17. The last bit "s" denotes the sign of level, "0" for positive and "1" for negative. The combinations of (LAST, RUN, LEVEL) not represented in these tables are decoded as described in subclause 7.4.1.3.

"

with

"

When short_video_header is 0, the variable length code table is different for intra blocks and inter blocks. The most commonly occurring EVENTs for the luminance and chrominance components of intra blocks in this case are decoded by referring to the intra columns of Table B-23 when reversible_vlc is set to '1' in I-, P-, or S(GMC)-VOPs, and by referring to Table B-16, otherwise. The most commonly occurring EVENTs for the luminance and chrominance components of inter blocks in this case are decoded by referring to the inter columns of Table B-23 when reversible_vlc is set to '1' in I-, P-, or S(GMC)-VOPs, and by referring to Table B-17, otherwise. The last bit "s" denotes the sign of level, "0" for positive and "1" for negative. The combinations of (LAST, RUN, LEVEL) not represented in these tables are decoded as described in subclause 7.4.1.3.

"

In subclause 7.4.1.3 Escape code, replace

"

Many possible EVENTS have no variable length code to represent them. In order to encode these statistically rare combinations an Escape Coding method is used. The escape codes of DCT coefficients are encoded in five modes. The first three of these modes are used when short_video_header is 0 and in the case that the reversible VLC tables are not used, and the fourth is used when short_video_header is 1. In the case that the reversible VLC tables are used, the fifth escape coding method as in Table B-23 is used. Their decoding process is specified below.

"

with

"

Many possible EVENTS have no variable length code to represent them. In order to encode these statistically rare combinations an Escape Coding method is used. The escape codes of DCT coefficients are encoded in five modes. The first three of these modes are used when short_video_header is 0 and in the case that the reversible VLC tables are not used, and the fourth is used when short_video_header is 1. In the case that the reversible VLC tables are used, the fifth escape coding method as in Table B-23 is used. Use of escape sequence of the reversible VLC (Table B-24 and Table B-25) for encoding the combinations listed in Table B-23 is prohibited. Their decoding process is specified below.

"

In subclause 7.4.4.6 Summary of quantiser process for method 1, replace

"

```
for (v=0; v<8;v++) {
    for (u=0; u<8;u++) {
        if (QF[v][u] == 0)
            F''[v][u] = 0;
        else if ( (u==0) && (v==0) && (macroblock_intra) ) {
            F''[v][u] = dc_scaler * QF[v][u];
        } else {
            if ( macroblock_intra ) {
                F''[v][u] = ( QF[v][u] * W[0][v][u] * quantiser_scale * 2 ) / 32;
            } else {
                F''[v][u] = ( ( ( QF[v][u] *  2 ) + Sign(QF[v][u]) ) * W[1][v][u]
                                                        * quantiser_scale ) / 32;
            }
        }
    }
}
```


"

with

"

```
for (v=0; v<8;v++) {
    for (u=0; u<8;u++) {
        if (QF[v][u] == 0)
            F''[v][u] = 0;
        else if ( (u==0) && (v==0) && (macroblock_intra) ) {
            F''[v][u] = dc_scaler * QF[v][u];
        } else {
```

```
                    if ( macroblock_intra ) {
                        F''[v][u] = ( QF[v][u] * W[0][v][u] * quantiser_scale * 2 ) / 16;
                    } else {
                        F''[v][u] = ( ( ( QF[v][u] *  2 ) + Sign(QF[v][u]) ) * W[1][v][u]
                                                                * quantiser_scale ) / 16;
                    }
                }
            }
        }
```

"

In subclause 7.5.2.1.2 P- and B-, and S(GMC)-VOPs, replace

"

The decoding of the current bab_type is dependent on the bab_type of the co-located bab in the reference VOP. The reference VOP is either a forward reference VOP or a backward reference VOP. The forward reference VOP is defined as the most recent non-empty (i.e. vop_coded != 0) I- or P-, or S(GMC)-VOP in the past, while the backward VOP is defined as the most recently decoded I- or P-, or S(GMC)-VOP in the future. If the current VOP is a P-, or S(GMC)-VOP, the forward reference VOP is selected as the reference VOP. If the current VOP is a B-VOP the following decision rules are applied:

1. If one of the reference VOPs is empty, the non-empty one (forward/backward) is selected as the reference VOP for the current B-VOP.

2. If both reference VOPs are non-empty, the forward reference VOP is selected if its temporal distance to the current B-VOP is not larger than that of the backward reference VOP, otherwise, the backward one is chosen.

"

with

"

The decoding of the current bab_type is dependent on the bab_type of the co-located bab in the reference VOP. The reference VOP is either a forward reference VOP or a backward reference VOP. The forward reference VOP is defined as the most recent non-empty (i.e. vop_coded != 0) I- or P-, or S(GMC)-VOP in the past, while the backward VOP is defined as the most recently decoded I- or P-, or S(GMC)-VOP in the future. If the current VOP is a P-, or S(GMC)-VOP, the forward reference VOP is selected as the reference VOP. If the current VOP is a B-VOP the following decision rules are applied:

1. If the backwards reference VOPs is empty, the non-empty one (forward) is selected as the reference VOP for the current B-VOP.

2. If both reference VOPs are non-empty, the forward reference VOP is selected if its temporal distance to the current B-VOP is not larger than that of the backward reference VOP, otherwise, the backward one is chosen.

"

In subclause 7.5.2.4 Motion compensation, replace

"

For inter mode babs (bab_type = 0,1,5 or 6), motion compensation is carried out by simple MV displacement according to the MVs.

Specifically, when bab_type is equal to 0 or 1 i.e. for the no-update modes, a displaced block of 16x16 pixels is copied from the binary alpha map of the previously decoded I or P-, or S(GMC)- VOP for which vop_coded is not equal to '0'. When the bab_type is equal to 5 or 6 i.e. when interCAE decoding is required, then the pixels immediately bordering the displaced block (to the left, right, top and bottom) are also copied from the most recent

valid reference VOP's (as defined in subclause 6.3.5**)** binary alpha map into a temporary shape block of 18x18 pixels size (see Figure 7-12). If the displaced position is outside the bounding rectangle, then these pixels are assumed to be "transparent".

If the current VOP is a B-VOP the following decision rules are applied:

- If one of the reference VOPs is empty (i.e. VOP_coded is 0), the non-empty one (forward/backward) is selected as the reference VOP for the current B-VOP.

- If both reference VOPs are non-empty, the forward reference VOP is selected if its temporal distance to the current B-VOP is not larger than that of the backward reference VOP, otherwise, the backward one is chosen.

"

with

"

For inter mode babs (bab_type = 0,1,5 or 6), motion compensation is carried out by simple MV displacement according to the MVs.

Specifically, when bab_type is equal to 0 or 1 i.e. for the no-update modes, a displaced block of 16x16 pixels is copied from the binary alpha map of the previously decoded I or P-, or S(GMC)- VOP for which vop_coded is not equal to '0'. When the bab_type is equal to 5 or 6 i.e. when interCAE decoding is required, then the pixels immediately bordering the displaced block (to the left, right, top and bottom) are also copied from the most recent valid reference VOP's (as defined in subclause 6.3.5**)** binary alpha map into a temporary shape block of 18x18 pixels size (see Figure 7-12). If the displaced position is outside the bounding rectangle, then these pixels are assumed to be "transparent".

If the current VOP is a B-VOP the following decision rules are applied:

- If the backwards reference VOPs is empty, the non-empty one (forward) is selected as the reference VOP for the current B-VOP.

- If both reference VOPs are non-empty, the forward reference VOP is selected if its temporal distance to the current B-VOP is not larger than that of the backward reference VOP, otherwise, the backward one is chosen.

"

Replace subclause 7.5.4.2 Decoding of enhancement layer with

"

7.5.4.2 Decoding of enhancement layer

When spatial scalability is enabled (scalability is set to 1 and hierarchy_type is set to 0) with enhancement_type == 0 or When spatial scalability is enabled with enhancement_type == 1 and use_ref_shape == 0, scalable shape coding process is used for decoding of binary shape.

If spatial scalability is enabled, use_ref_shape is set to 1 and enhancement_type is set to 1, the same non-scalable decoding process is applied for binary shape of enhancement layer. In this case, the following rules are applied for enhancement layer.

1. In PVOP, Inter shape coding should be done as bab_type of co-located MB in the reference VOP (lower layer) is "Opaque".

2. In BVOP, forward reference VOP, most recently decoded non-empty VOP in the same layer, is always selected as reference VOP of shape coding.