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)**

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.

If spatial scalability is enabled, use_ref_shape is set to 1 and enhancement_type is set to 0, then the up-sampled binary shape from reference layer is used for the binary shape of enhancement layer. The up sampling and down sampling process of this purpose also follows up-down sampling method described in the subclause 7.5.4.4.

When spatial scalability is enabled and enhancement_type is set to 0, in the enhancement layer, the forward prediction in P-VOP and the backward prediction in B-VOP are used as the spatial prediction. In that case the shape information is coded by scan interleaving (SI) based method. For the forward prediction in B-VOP a motion compensated temporal prediction is made from the reference VOP in the enhancement layer. In that case the shape information is coded by the CAE method as like in base layer except that the shape motion vectors are obtained from those of the collocated bab in the lower layer. Motion vector and shape coding mode(bab_type) of collocated bab in the lower layer are used for decoding the enhancement layer bab.

The location of collocated bab in the lower layer can be found by following method.

collocated_MBX
= min (max (0, current_MBX*shape_hor_sampling_factor_m/shape_hor_sampling_factor_n), NumMBXLower-1);

collocated_MBY
= min (max(0, current_MBY*shape_ver_sampling_factor_m/shape_ver_sampling_factor_n), NumMBYLower-1);

For the current MB location [current_MBX, current_MBY], the location of collocated bab in the reference layer is denoted as [collocated_MBX, collocated_MBY]. current_MBX, current_MBY, collocated_MBX and collocated_MBY are the MB-unit coordinations. NumMBXLower and NumMBYLower denote the number of micro-blocks in the lower layer VOP on horizontal and vertical directions, respectively.

"

In subclause 7.5.4.4 Spatial prediction, replace

"

The spatial prediction is made by resampling the lower reference layer reconstructed VOP to the same sampling grid as the enhancement layer. For the resampling, repetition is used on the the lower layer.

For enhancement layer encoding/decoding, the base layer VOP should be up-sampled as the sampling ratio, which is included in the VOL syntax. In VOL syntax for enhancement layer, there are four fields for the up-sampling ratio, i.e., shape_hor_sampling_factor_n, shape_hor_sampling_factor_m, shape_vert_sampling_factor_n and shape_vert_sampling_factor_n.

"

with

"

The spatial prediction is made by resampling the lower reference layer reconstructed VOP to the same sampling grid as the enhancement layer. For the resampling, repetition is used on the lower layer.

For enhancement layer encoding/decoding, the reference layer VOP should be up-sampled as the sampling ratio, which is included in the VOL syntax. In VOL syntax for enhancement layer, there are four fields for the up-sampling ratio, i.e., shape_hor_sampling_factor_n, shape_hor_sampling_factor_m, shape_vert_sampling_factor_n and shape_vert_sampling_factor_n.

"

In subclause 7.5.4.8 Intra coded enhancement layer decoding, replace

"

Intra coded enhancement layer decoding uses scan interleaving algorithm before performing intra-mode CAE. The decoding order with SI scanning is as follows:

1. Copy B from base layer

2. Decoding order with Vertical scanning : Vr --> Vp1 --> Vp2 --> ... --> Vpk

3. Decoding order with Horizontal scanning : Hr --> Hp1 --> Hp2 --> ... --> Hpl

where,

B : Piexl that can be copied from collocated pixel in the base layer.

"

with

"

Intra coded enhancement layer decoding uses scan interleaving algorithm before performing intra-mode CAE. The decoding order with SI scanning is as follows:

1. Copy B from reference layer

2. Decoding order with Vertical scanning : Vr --> Vp1 --> Vp2 --> ... --> Vpk

3. Decoding order with Horizontal scanning : Hr --> Hp1 --> Hp2 --> ... --> Hpl

where,

B : Piexl that can be copied from collocated pixel in the reference layer.

"

In subclause 7.6.3, replace

"

if(quarter_pel==1)

”

with

“

if(quarter_sample==1)

”

In subclause 7.6.3, add immediately following the pseudocode

"

The value of mv_data (i.e., horizontal_mv_data and vertical_mv_data) is equal to two times the value found in the 'vector differences' column of Table B-12 associated with the received codeword.

"

In subclause 7.6.3, add a footnote attached to the last entry of Table B-12

"

The last code shall not be used when vop_fcode=1.

"

In subclause 7.6.3, replace

"

The parameters in the bitstream shall be such that the components of the reconstructed differential motion vector, MVDx and MVDy, shall lie in the range [low:high].

"

with

"

The parameters in the bitstream shall be such that the components of the reconstructed differential motion vector, MVDx and MVDy, shall lie in the range [low:high], at the time of their use in calculating the values of MVx and MVy (i.e., intermediate values of MVDx and MVDy may occur that are outside the range [low:high]).

"

Replace subclause 7.8.5 Warping with

"

7.8.5 Warping

For any pixel (i, j) inside the VOP boundary, $(F(i, j), G(i, j))$ and $(F_c(i_c, j_c), G_c(i_c, j_c))$ are computed as described in the following. These quantities are then used for sample reconstruction as specified in subclause 7.8.6. The following notations are used to simplify the description:

$$I = i - i_0,$$
$$J = j - j_0,$$
$$I_c = 4\, i_c - 2\, i_0 + 1,$$
$$J_c = 4\, j_c - 2\, j_0 + 1,$$

When no_of_sprite_warping_point == 0,

$(F(i, j), G(i, j)) = (s\, i, s\, j)$,
$(F_c(i_c, j_c), G_c(i_c, j_c)) = (s\, i_c, s\, j_c)$.

When no_of_sprite_warping_point == 1 and sprite_enable == 'static,

$(F(i, j), G(i, j)) = (i_0' + sI, j_0' + s\, J)$,
$(F_c(i_c, j_c), G_c(i_c, j_c)) = i_0' /// 2 + s\,(i_c - i_0 / 2),\ j_0' /// 2 + s\,(j_c - j_0 / 2))$.

When no_of_sprite_warping_point == 1 and sprite_enable == 'GMC',

$(F(i, j), G(i, j)) = (i_0' + sI, j_0' + s\, J)$,
$(F_c(i_c, j_c), G_c(i_c, j_c)) = (((i_0' >> 1) \mid (i_0' \,\&\, 1)) + s\,(i_c - i_0 / 2),\ ((j_0' >> 1) \mid (j_0' \,\&\, 1)) + s\,(j_c - j_0 / 2))$

When no_of_sprite_warping_points == 2,

$$
\begin{aligned}
(F(i, j), G(i, j)) &= (i_0' + ((-r\, i_0' + i_1'')\, I + (r\, j_0' - j_1'')\, J) /// (W'\, r)\,, \\
&\quad\ j_0' + ((-r\, j_0' + j_1'')\, I + (-r\, i_0' + i_1'')\, J) /// (W'\, r)), \\
(F_c(i_c, j_c), G_c(i_c, j_c)) &= (((-r\, i_0' + i_1'')\, I_c + (r\, j_0' - j_1'')\, J_c + 2\, W'\, r\, i_0' - 16W') /// (4\, W'\, r), \\
&\quad\ ((-r\, j_0' + j_1'')\, I_c + (-r\, i_0' + i_1'')\, J_c + 2\, W'\, r\, j_0' - 16W') /// (4\, W'\, r)).
\end{aligned}
$$

According to the definition of W' and H' (i.e. $W' = 2^{\alpha}$ and $H' = 2^{\beta}$), the divisions by "///" in these functions can be replaced by binary shift operations. By this replacement, the above equations can be rewritten as:

$$
\begin{aligned}
(F(i, j), G(i, j)) &= (i_0' + (((-r\, i_0' + i_1'')\, I + (r\, j_0' - j_1'')\, J + 2^{\alpha+\rho-1}) >> (\alpha+\rho))\,, \\
&\quad\ j_0' + (((-r\, j_0' + j_1'')\, I + (-r\, i_0' + i_1'')\, J + 2^{\alpha+\rho-1}) >> (\alpha+\rho)), \\
(F_c(i_c, j_c), G_c(i_c, j_c)) &= (((-r\, i_0' + i_1'')\, I_c + (r\, j_0' - j_1'')\, J_c + 2\, W'\, r\, i_0' - 16W' + 2^{\alpha+\rho+1}) >> (\alpha+\rho+2), \\
&\quad\ ((-r\, j_0' + j_1'')\, I_c + (-r\, i_0' + i_1'')\, J_c + 2\, W'\, r\, j_0' - 16W' + 2^{\alpha+\rho+1}) >> (\alpha+\rho+2)),
\end{aligned}
$$

where $2^{\rho}=r$.

When no_of_sprite_warping_points == 3,

$$
\begin{aligned}
(F(i, j), G(i, j)) &= (i_0' + ((-r\, i_0' + i_1'')\, H'\, I + (-r\, i_0' + i_2'')W'\, J) /// (W'H'r), \\
&\quad\ j_0' + ((-r\, j_0' + j_1'')\, H'\, I + (-r\, j_0' + j_2'')W'\, J) /// (W'H'r)), \\
(F_c(i_c, j_c), G_c(i_c, j_c)) &= (((-r\, i_0' + i_1'')\, H'\, I_c + (-r\, i_0' + i_2'')W'\, J_c + 2\, W'H'r\, i_0' - 16W'H') /// (4W'H'r), \\
&\quad\ ((-r\, j_0' + j_1'')\, H'\, I_c + (-r\, j_0' + j_2'')W'\, J_c + 2\, W'H'r\, j_0' - 16W'H') /// (4W'H'r)).
\end{aligned}
$$

According to the definition of W' and H', the computation of these functions can be simplified by dividing the denominator and numerator of division beforehand by W' (when $W' < H'$) or H' (when $W' \geq H'$). As in the case of no_of_sprite_warping_points == 2, the divisions by "///" in these functions can be replaced by binary shift operations. For example, when $W' \geq H'$ (i.e. $\alpha \geq \beta$) the above equations can be rewritten as:

$$
\begin{aligned}
(F(i, j), G(i, j)) &= (i_0' + (((-r\, i_0' + i_1'')\, I + (-r\, i_0' + i_2'')\, 2^{\alpha-\beta} J + 2^{\alpha+\rho-1}) >> (\alpha+\rho)), \\
&\quad\ j_0' + (((-r\, j_0' + j_1'')\, I + (-r\, j_0' + j_2'')\, 2^{\alpha-\beta} J + 2^{\alpha+\rho-1}) >> (\alpha+\rho))), \\
(F_c(i_c, j_c), G_c(i_c, j_c)) &= (((-r\, i_0' + i_1'')\, I_c + (-r\, i_0' + i_2'')\, 2^{\alpha-\beta} J_c + 2W'r\, i_0' - 16W' + 2^{\alpha+\rho+1}) >> (\alpha+\rho+2), \\
&\quad\ ((-r\, j_0' + j_1'')\, I_c + (-r\, j_0' + j_2'')\, 2^{\alpha-\beta} J_c + 2W'r\, j_0' - 16W' + 2^{\alpha+\rho+1}) >> (\alpha+\rho+2)).
\end{aligned}
$$

When no_of_sprite_warping_point == 4,

$$
\begin{aligned}
(F(i, j), G(i, j)) &= ((a\, I + b\, J + c) /// (g\, I + h\, J + D\, W\, H), \\
&\quad\ (d\, I + e\, J + f) /// (g\, I + h\, J + D\, W\, H)), \\
(F_c(i_c, j_c), G_c(i_c, j_c)) &= ((2\, a\, I_c + 2\, b\, J_c + 4\, c - (g\, I_c + h\, J_c + 2\, D\, W\, H)\, s) /// (4gI_c + 4\, hJ_c + 8D\, W\, H), \\
&\quad\ (2\, d\, I_c + 2\, e\, J_c + 4\, f - (g\, I_c + h\, J_c + 2\, D\, W\, H)\, s) /// (4\, g\, I_c + 4\, hJ_c + 8D\, W\, H))
\end{aligned}
$$

where

$$
\begin{aligned}
g &= ((i_0' - i_1' - i_2' + i_3')\, (j_2' - j_3') - (i_2' - i_3')\, (j_0' - j_1' - j_2' + j_3'))\, H\,, \\
h &= ((i_1' - i_3')\, (j_0' - j_1' - j_2' + j_3') - (i_0' - i_1' - i_2' + i_3')\, (j_1' - j_3'))\, W\,, \\
D &= (i_1' - i_3')\, (j_2' - j_3') - (i_2' - i_3')\, (j_1' - j_3'), \\
a &= D\, (i_1' - i_0')\, H + g\, i_1', \\
b &= D\, (i_2' - i_0')\, W + h\, i_2', \\
c &= D\, i_0'\, W\, H, \\
d &= D\, (j_1' - j_0')\, H + g\, j_1', \\
e &= D\, (j_2' - j_0')\, W + h\, j_2', \\
f &= D\, j_0'\, W\, H.
\end{aligned}
$$

A set of parameters that causes the denominator of any of the above equations to be zero for any pixel in a opaque or boundary macroblock is disallowed. The implementor should be aware that a 32bit register may not be sufficient for representing the denominator or the numerator in the above transform functions for affine and perspective transform. The usage of a 64 bit floating point representation should be sufficient in such case.

"

Replace subclause 7.9 Generalized scalable decoding with

"

7.9 Generalized scalable decoding

This subclause specifies the additional decoding process required for decoding scalable coded video.

The scalability framework is referred to as generalized scalability which includes the spatial and the temporal scalabilities. The temporal scalability offers scalability of the temporal resolution, and the spatial scalability offers scalability of the spatial resolution. Each type of scalability involves more than one layer. In the case of two layers, consisting of a lower layer and a higher layer; the lower layer is referred to as the reference layer and the higher layer is called the enhancement layer.

In the case of temporal scalability, both rectangular VOPs as well as arbitrary shaped VOPs are supported. In the case of spatial scalability, only rectangular VOPs are supported. Figure 7-32 shows a high level decoder structure for generalized scalability.

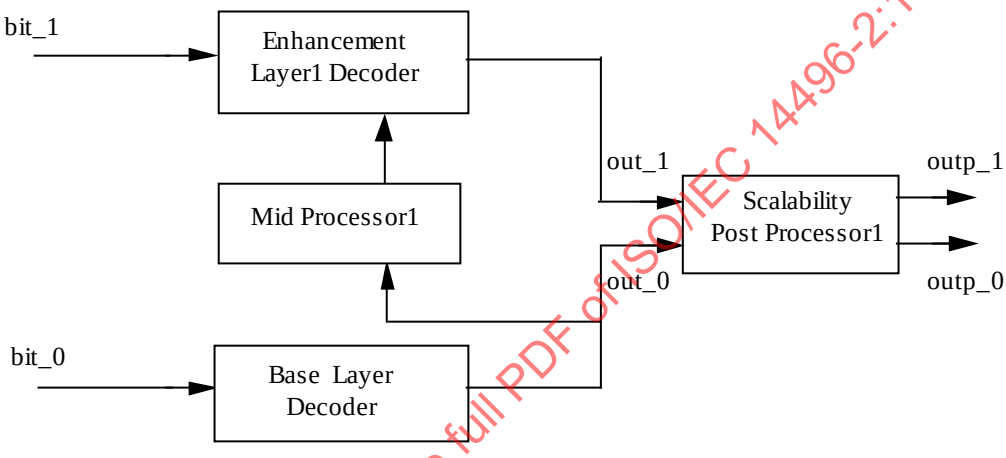

Figure 7-32 -- High level decoder structure for generalized scalability

The base layer and enhancement layer bitstreams are input for decoding by the corresponding base layer decoder and enhancement layer decoder.

When spatial scalability is to be performed, mid processor 1 performs spatial up or down sampling of input. The scalability post processor performs any necessary operations such as spatial up or down sampling of the decoded reference layer for display resulting at outp_0 while the enhancement layer without resolution conversion may be output as outp_1.

When temporal scalability is to be performed, the decoding of base and enhancement layer bitstreams occurs in the corresponding base and enhancement layer decoders as shown. In this case, mid processor 1 does not perform any spatial resolution conversion. The post processor simply outputs the base layer VOPs without any conversion, but temporally multiplexes the base and enhancement layer VOPs to produce higher temporal resolution enhancement layer.

The reference VOPs for prediction are selected by ref_select_code as specified in Table 7-13 and Table 7-14. In coding of P-VOPs belonging to an enhancement layer, the forward reference is one of the following four: the most recently decoded VOP of enhancement layer, the most recent VOP of the reference layer in display order, the next VOP of the reference layer in display order, or the temporally coincident VOP in the reference layer.

In B-VOPs, the forward reference is one of the following two: the most recently decoded enhancement VOP or the most recent reference layer VOP in display order. The backward reference is one of the following three: the temporally coincident VOP in the reference layer, the most recent reference layer VOP in display order, or the next reference layer VOP in display order.

In case of hierarchy_type == 1(temporal scalability), the default value of ref_select_code is “00”. The value of vop_coded should be “1” for the first VOP in an enhancement layer if the value of video_object_layer_shape is ‘00’.

In case of hierarchy_type == 0(spatial scalability), the default value of ref_select_code is “11” for P-VOP and "00" for B-VOP. The value of vop_coded should be “1” for the first VOP in an enhancement layer if the value of video_object_layer_shape is ‘00’.

Table 7-13 -- Prediction reference choices in enhancement layer P-VOPs for scalability

ref_select_code	forward prediction reference
00	Most recently decoded enhancement VOP belonging to the same layer for which “vop_coded == 1”.
01	Most recently VOP in display order belonging to the reference layer for which “vop_coded == 1”.
10	Next VOP in display order belonging to the reference layer.
11	Temporally coincident VOP in the reference layer (no motion vectors)

NOTE 1 — The value “10” is prohibited if “vop_coded = 0” for the next VOP in display order belonging to the reference layer.

NOTE 2 — The value “11” is prohibited if “vop_coded = 0” for the temporally coincident VOP in the reference layer.

Table 7-14 -- Prediction reference choices in enhancement layer B-VOPs for scalability

ref_select_code	forward temporal reference	backward temporal reference
00	Most recently decoded enhancement VOP of the same layer for which “vop_coded == 1”.	Temporally coincident VOP in the reference layer (no motion vectors)
01	Most recently decoded enhancement VOP of the same layer for which “vop_coded == 1”.	Most recent VOP in display order belonging to the reference layer.
10	Most recently decoded enhancement VOP of the same layer for which “vop_coded == 1”.	Next VOP in display order belonging to the reference layer.
11	Most recently VOP in display order belonging to the reference layer for which “vop_coded == 1”.	Next VOP in display order belonging to the reference layer.

The following combinations of video_object_layer_shape in the base and enhancement layer and enhancement_type in the enhancement_layer are supported.

	Base Layer	Enhancement Layer	Enhancement _type	status
hiearachy_type=1	rectangular	rectangular	0	supported
	rectangular	binary shape	1	supported
	binary Shape	binary shape	0/1	supported
	binary Shape	rectangular	-	forbidden
	binary only	binary only	-	forbidden
hiearachy_type=0 && video_object_layer_ver_id = 0001	rectangular	rectangular	0	supported
	rectangular	binary shape	-	forbidden
	binary shape	binary shape	-	forbidden
	binary shape	rectangular	-	forbidden
	binary only	binary only	-	forbidden
hiearachy_type=0 && video_object_layer_ver_id = 0010	rectangular	rectangular	0	supported
	rectangular	binary shape	1	supported
	binary shape	binary shape	0/1	supported
	binary shape	rectangular	-	forbidden
	binary only	binary only	-	supported

"

In subclause 7.9.1.1 Base layer and enhancement layer, replace

"

In the case of temporal scalability, the decoded VOPs of the enhancement layer are used to increase the frame rate of the base layer. Figure 7-33 shows a simplified diagram of the motion compensation process for the enhancement layer using temporal scalability.

Predicted samples p[y][x] are formed either from frame stores of base layer or from frame stores of enhancement layer. The difference data samples f[y][x] are added to p[y][x] to form the decoded samples d[y][x].

There are two types of enhancement structures indicated by the “enhancement_type” flag. When the value of enhancement_type is “1”, the enhancement layer increases the temporal resolution of a partial region of the base layer. When the value of enhancement_type is “0”, the enhancement layer increases the temporal resolution of an entire region of the base layer.

"

with

"

In the case of temporal scalability, the decoded VOPs of the enhancement layer are used to increase the frame rate of the base layer. Figure 7-33 shows a simplified diagram of the motion compensation process for the enhancement layer using temporal scalability.

Predicted samples p[y][x] are formed either from frame stores of reference layer or from frame stores of enhancement layer. The difference data samples f[y][x] are added to p[y][x] to form the decoded samples d[y][x].

There are two types of enhancement structures indicated by the “enhancement_type” flag. When the value of enhancement_type is “1”, the enhancement layer increases the temporal resolution of a partial region of the reference layer. When the value of enhancement_type is “0”, the enhancement layer increases the temporal resolution of an entire region of the reference layer.

The value of video_object_layer_width in base layer and enhancement layer shall be identical if both layers are rectangular. The value of video_object_layer_height in base layer and enhancement layer shall be identical if both layers are rectangular

"

Replace subclause 7.9.1.3 Enhancement layer with

"

7.9.1.3 Enhancement layer

The VOP of the enhancement layer is decoded as either I-VOP, P-VOP or B-VOP. The shape of the VOP is either rectangular (video_object_layer_shape is “00”) or arbitrary (video_object_layer_shape is “01”). B-VOP in base layer shall not be used as a reference for enhancement layer VOP although B-VOP in enhancement layer can be a reference for enhancement layer VOP.

"

Replace subclause 7.9.1.3.1 Decoding of I-VOPs with

"

7.9.1.3.1 Decoding of I-VOPs

The decoding process of the I-VOPs in the enhancement layer is the same as the non-scalable decoding process. ref_layer_id, ref_layer_sampling_direc, hor_sampling_factor_n, horsampling_factor_m, vertical_sampling_factor_n , vertical_sampling_factor_m and ref_select_code are ignored in the temporal scalability I-VOPs.

"

Replace subclause 7.9.1.3.2 Decoding of P-VOPs with

"

7.9.1.3.2 Decoding of P-VOPs

The reference layer is indicated by ref_layer_id in Video Object Layer class. Other decoding process is the same as non-scalable P-VOPs except the process specified in subclauses 7.9.1.3.4 and 7.9.1.3.5.

For P-VOPs, the ref_select_code is either “00”, “01” or “10”.

When the value of ref_select_code is “00”, the prediction reference is set by the most recently decoded VOP belonging to the same layer.

When the value of ref_select_code is “01”, the prediction reference is set by the previous VOP in display order belonging to the reference layer.

When the value of ref_select_code is “10”, the prediction reference is set by the next VOP in display order belonging to the reference layer

In the case of hierarchy_type=1(temporal scalability), hor_sampling_factor_n, hor_sampling_factor_m, vert_sampling_factor_n and vert_sampling_factor_m are ignored.

"

Replace subclause 7.9.1.3.3 Decoding of B-VOPs with

"

7.9.1.3.3 Decoding of B-VOPs

The reference layer is indicated by ref_layer_id in Video Object Layer class. Other decoding process is the same as non-scalable B-VOPs except the process specified in subclauses 7.9.1.3.4 and 7.9.1.3.5.

For B-VOPs, the ref_select_code is either “01”, “10” or “11”.

When the value of ref_select_code is “01”, the forward prediction reference is set by the most recently decoded VOP belonging to the same layer and the backward prediction reference is set by the previous VOP in display order belonging to the reference layer.

When the value of ref_select_code is “10”, the forward prediction reference is set by the most recently decoded VOP belonging to the same layer, and the backward prediction reference is set by the next VOP in display order belonging to the reference layer.

When the value of ref_select_code is “11”, the forward prediction reference is set by the previous VOP in display order belonging to the reference layer and the backward prediction reference is set by the next VOP in display order belonging to the reference layer. The picture type of the reference VOP shall be either I or P (vop_coding_type = “00” or “01”).

When the value of ref_select_code is “01” or “10”, direct mode is not allowed. modb shall always exist in each macroblock, i.e. the macroblock is not skipped even if the co-located macroblock is skipped.

In the case of hierarchy_type=1(temporal scalability), hor_sampling_factor_n, hor_sampling_factor_m, vert_sampling_factor_n and vert_sampling_factor_m are ignored.

"

In subclause 7.9.2.3 Base layer and enhancement layer, replace

"

In the case of spatial scalability, the enhancement bitstream is used to increase the resolution of the image. When the output with lower resolution is required, only the base layer is decoded. When the output with higher resolution is required, both the base layer and the enhancement layer are decoded.

"

with

"

In the case of spatial scalability, the enhancement bitstream is used to increase the resolution of the image. When the output with the lowest resolution is required, only the base layer is decoded. When the output with higher resolution is required, both the base layer and the enhancement layer are decoded.

"

In subclause 7.9.2.5 Prediction in the enhancement layer, replace

"

A motion compensated temporal prediction is made from reference VOPs in the enhancement layer. In addition, a spatial prediction is formed from the lower layer decoded frame ($d_{lower}[y][x]$). These predictions are selected individually or combined to form the actual prediction.

In the enhancement layer, the forward prediction in P-VOP and the backward prediction in B-VOP are used as the spatial prediction. The reference VOP is set to the temporally coincident VOP in the base layer. The forward prediction in B-VOP is used as the temporal prediction from the enhancement layer VOP. The reference VOP is set to the most recently decoded VOP of the enhancement layer. The interpolate prediction in B-VOP is the combination of these predictions. If the reference VOP has arbitrary shape, padding process shall be done before resampling.

"