INTERNATIONAL STANDARD

ISO/IEC 14496-2

Second edition
2001-12-01

AMENDMENT 2
2002-02-01

Information technology — Coding of audio-visual objects —

Part 2: Visual

AMENDMENT 2: Streaming video profile

Technologies de l'information — Codage des objets audiovisuels —

Partie 2: Codage visuel

AMENDEMENT 2: Cours du profil vidéo

Reference number
ISO/IEC 14496-2:2001/Amd.2:2002(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

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 3.

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 Amendment 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 International Standard ISO/IEC 14496-2:2001 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects — Part 2: Visual

AMENDMENT 2: Streaming video profile

1) Add the following text to the end of ‘Purpose’ of ‘Introduction’:

“

Two profiles are developed in response to the growing need for a video coding method for Streaming Video on Internet applications. It provides the definition and description of Advanced Simple (AS) Profile and Fine Granularity Scalable (FGS) Profile. AS Profile provides the capability to distribute single-layer frame based video at a wide range of bit rates available for the distribution of video on Internet. FGS Profile uses AS Video Object in the base layer and provides the description of two enhancement layer types - Fine Granularity Scalability (FGS) and FGS Temporal Scalability (FGST). FGS Profile allows the coverage of a wide range of bit rates for the distribution of video on Internet with the flexibility of using multiple layers, where there is a wide range of bandwidth variation.

“

2) Add the following text into ‘Introduction’ following ‘Error Resilience’:

“

Fine Granularity Scalability

Fine Granularity Scalability (FGS) provides quality scalability for each VOP. Figure AMD2-1 shows a basic FGS decoder structure.

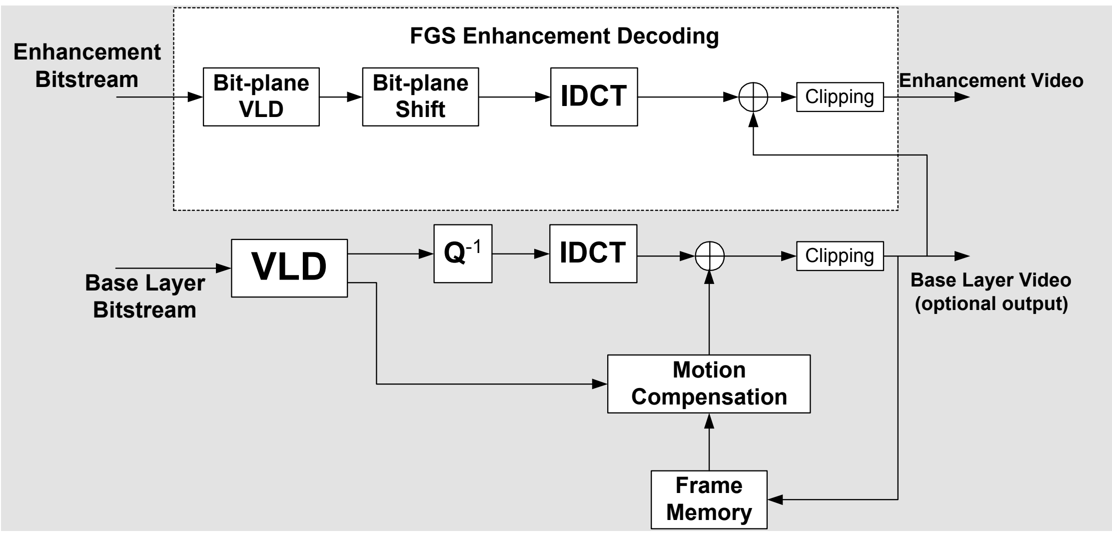

Figure AMD2-1 — A Basic FGS Decoder Structure

To reconstruct the enhanced VOP, the enhancement bitstream is first decoded using bit-plane VLD. The decoded block-bps are used to reconstruct the DCT coefficients in the DCT domain which are then right-shifted based on the frequency weighting and selective enhancement shifting factors. The output of bit-plane shift is the DCT coefficients of the image domain residues. After the IDCT, the image domain residues are reconstructed. They are added to the reconstructed clipped base-layer pixels to reconstruct the enhanced VOP. The reconstructed enhanced VOP pixels are limited into the value range between 0 and 255 by the clipping unit in the enhancement layer to generate the final enhanced video. The reconstructed base layer video is available as an optional output since each base layer reconstructed VOP needs to be stored in the frame buffer for motion compensation.

The basic FGS enhancement layer consists of FGS VOPs that enhance the quality of the base-layer VOPs as shown in Figure AMD2-2.

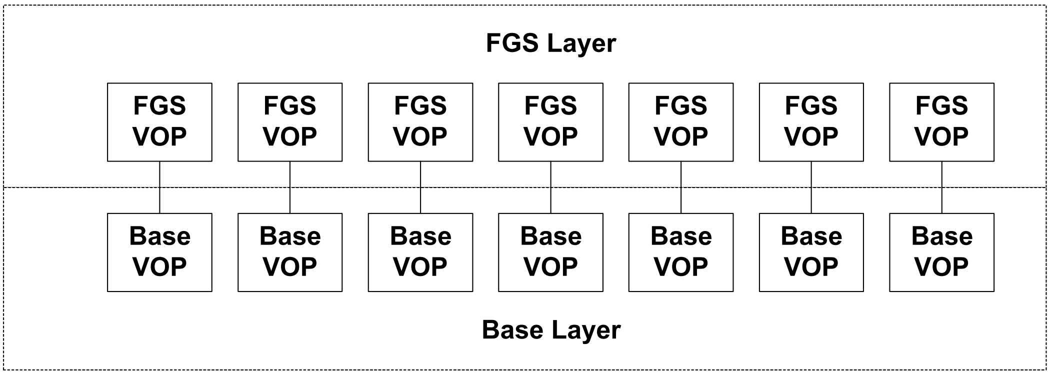

Figure AMD2-2 — Basic FGS Enhancement Structure

When FGS temporal scalability (FGST) is used, there are two possible enhancement structures. One structure is to have two separate enhancement layers for FGS and FGST as shown in Figure AMD2-3 and the other structure is to have one combined enhancement layer for FGS and FGST as shown in Figure AMD2-4.

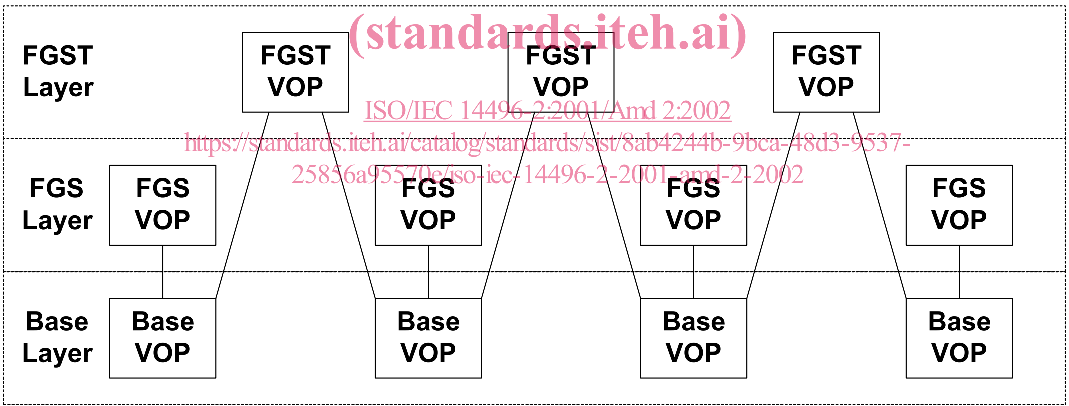

Figure AMD2-3 — Two Separate Enhancement Layers for FGS and FGST

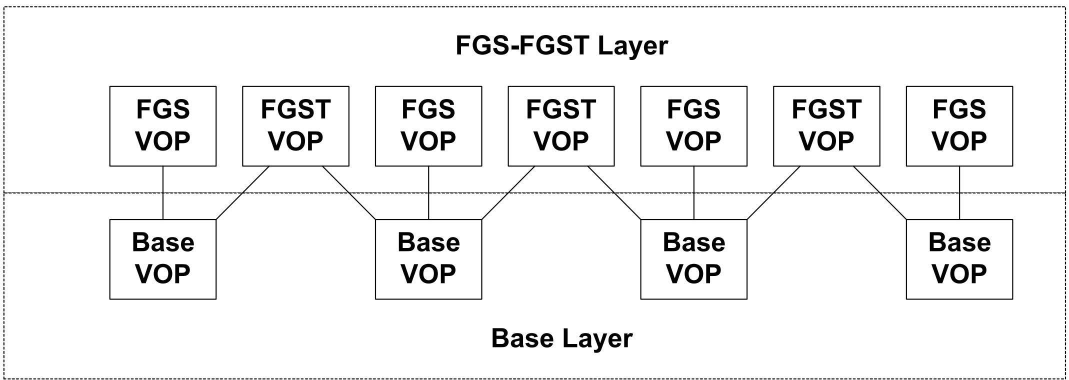

Figure AMD2-4 — One Combined Enhancement Layer for FGS and FGST

In either one of these two structures that include FGS temporal scalability, the prediction for the FGS temporal scalable VOPs can only be from the base layer. Each FGS temporal scalable VOP has two separate parts. The first part contains motion vector data and the second part contains the DCT texture data. The syntax for the first part is similar to that in the temporal scalability described in subclause 6.2. The DCT texture data in the second part are coded using bit-plane coding in the same way as that in FGS. To distinguish the temporal scalability in subclause 6.2 and FGS temporal scalability, the FGS temporal scalability layer in Figure AMD2-3 is called “FGST layer”. The combined FGS and FGST layer in Figure AMD2-4 is called “FGS-FGST layer”. The “FGS VOP” shown in Figure AMD2-3 and Figure AMD2-4 is an fgs vop with **fgs_vop_coding_type** being ‘I’. The “FGST VOP” shown in Figure AMD2-3 and Figure AMD2-4 is an fgs vop with **fgs_vop_coding_type** being ‘P’ or ‘B’.

The code value of **profile_and_level_indication** in VisualObjectSequence() has been extended to include the profile and level indications for AS Profile and FGS Profile. The identifier for an enhancement layer is the syntax **video_object_type_indication** in VideoObjectLayer(). A unique code is defined for FGS Object Type to indicate that this VOL contains fgs vops. Another unique code is defined for AS Object Type to indicate that this VOL is the base-layer. There is a syntax **fgs_layer_type** in VideoObjectLayer() to indicate whether this VOL is an FGS layer as shown in Figure AMD2-2 and Figure AMD2-3, or an FGST layer as shown in Figure AMD2-3, or an FGS-FGST layer as shown in Figure AMD2-4. Similar to the syntax structure in subclause 6.2, under each VOL for FGS, there is a hierarchy of fgs vop, fgs macroblock, and fgs block. An fgs vop starts with a unique **fgs_vop_start_code**. Within each fgs vop, there are multiple vop-bps. Each vop-bp in an fgs vop starts with an **fgs_bp_start_code** whose last 5 bits indicate the ID of the vop-bp. In each fgs macroblock, there are 4 block-bps for the luminance component (Y), 2 block-bps for the two chrominance components (U and V) for the 4:2:0 chrominance format. Each block-bp is coded by VLC.

“

3) Add the following subclauses in clause 3

"

3.AMD2.1 **block-bp**: An array of 64 bits, one from each DCT coefficient at the same significant position of accuracy in a zigzag scan order. When frequency weighting is used, block-bps are formed after the weighting is applied to the DCT coefficients in an 8x8 block.

3.AMD2.2 **end of plane; eop**: A symbol to indicate whether a ‘1’ bit is the last 1’ bit of a block-bp.

3.AMD2.3 **fgs block**: An 8-row by 8-column matrix of bits, each from one DCT coefficient at the same significant position of accuracy, or its coded representation. The usage is clear from the context.

3.AMD2.4 **fgs macroblock**: The four block-bps of luminance component (Y) and the two (for 4:2:0 chrominance format) corresponding block-bps of chrominance components (U and V) with the same accuracy significance coming from the DCT coefficients of a macroblock. It may also be used to refer to the coded representation of the six block-bps. The usage is clear from the context.

3.AMD2.5 **fgs macroblock number**: A number for an fgs macroblock within a vop-bp. The fgs macroblock number of the top-left fgs macroblock in each vop-bp shall be zero. The fgs macroblock number increments from left to right and from top to bottom.

3.AMD2.6 **fgs run**: The number of ‘0’ bits preceding a ‘1’ bit within a block-bp.

3.AMD2.7 **fgs temporal scalability; FGST**: A type of scalability where an enhancement layer uses predictions from sample data derived from the base layer using motion vectors. The VOP size in the enhancement layer is the same as that in the base layer. FGST is a specific type of temporal scalability where all DCT coefficients are coded using bit-plane coding as in FGS.

3.AMD2.8 **fgs vop**: The pixel differences between the original VOP and the reconstructed VOP in the base layer. It may be used to refer to the DCT coefficients of the pixel differences or the original VOP. It may also be used to refer to the coded representation of the DCT coefficients. In the context of FGST, fgs vop refers to the original temporal scalable VOP. The usage is clear from the context.

3.AMD2.9 **fine granularity scalability; FGS**: A type of scalability where an enhancement layer uses prediction from sample data of reconstructed VOP in the base layer. The encoded bitstream for each fgs vop can be truncated into any number of bits. The truncated bitstream for each fgs vop can be decoded to provide quality enhancement proportional to the amount of bits in the truncated bitstream of the fgs vop. The fgs vop has the same size and VOP rate as those of the base layer.

3.AMD2.10 **vop-bp**: An array of block-bps with the same accuracy significance in an fgs vop. There are three color components (Y, U, and V) in a vop-bp. Each color component in a vop-bp consists of all the block-bps of that color.

“

4) Add the following subclause to subclause 5.2:

“

Definition of start_of_bit_plane() function

The function start_of_bit_plane() returns 1 if the next bit in the bitstream is the first bit of the codes associated with a vop-bp. Otherwise it returns 0.

“

5) Add the following text to the end of subclause 6.1:

“

In a typical application of FGS, the bitstream at the input of an FGS decoder is a truncated version of the bitstream at the output of an FGS encoder. It is likely that, at the end of each fgs vop before the next fgs_vop_start_code, only partial bits of the fgs vop are at the input of the decoder due to truncation of the fgs vop bitstream. Decoding of the truncated bitstream is not normative. An example of dealing with the truncated bitstream is described in Annex S. The FGS syntax description in this clause is for a complete bitstream without truncation.

“

6) Replace Table 6-3 in subclause 6.2.1 with the following table:

“

Table 6-3. Start code values

name	**start code value (hexadecimal)**
video_object_start_code	00 through 1F
video_object_layer_start_code	20 through 2F
reserved	30 through 3F
fgs_bp_start_code	40 through 5F
reserved	60 through AF
visual_object_sequence_start_code	B0
visual_object_sequence_end_code	B1
user_data_start_code	B2
group_of_vop_start_code	B3
video_session_error_code	B4
visual_object_start_code	B5
vop_start_code	B6

slice_start_code	B7
extension_start_code	B8
fgs_vop_start_code	B9
fba_object_start_code	BA
fba_object_plane_start_code	BB
mesh_object_start_code	BC
mesh_object_plane_start_code	BD
still_texture_object_start_code	BE
texture_spatial_layer_start_code	BF
texture_snr_layer_start_code	C0
texture_tile_start_code	C1
texture_shape_layer_start_code	C2
reserved	C3-C5
System start codes (see note)	C6 through FF
NOTE System start codes are defined in ISO/IEC 14496-1	

"

7) Replace VideoObjectLayer() in subclause 6.2.3:

"

VideoObjectLayer() {	**No. of bits**	**Mnemonic**
if(next_bits() == video_object_layer_start_code) {		
short_video_header = 0		
video_object_layer_start_code	32	bslbf
random_accessible_vol	1	bslbf
video_object_type_indication	8	uimsbf
is_object_layer_identifier	1	uimsbf
if (is_object_layer_identifier) {		
video_object_layer_verid	4	uimsbf
video_object_layer_priority	3	uimsbf
}		
aspect_ratio_info	4	uimsbf
if (aspect_ratio_info == "extended_PAR") {		
par_width	8	uimsbf
par_height	8	uimsbf
}		
vol_control_parameters	1	bslbf
if (vol_control_parameters) {		
chroma_format	2	uimsbf
low_delay	1	uimsbf
vbv_parameters	1	blsbf
if (vbv_parameters) {		
first_half_bit_rate	15	uimsbf
marker_bit	1	bslbf
latter_half_bit_rate	15	uimsbf
marker_bit	1	bslbf
first_half_vbv_buffer_size	15	uimsbf

marker_bit	1	bslbf
latter_half_vbv_buffer_size	3	uimsbf
first_half_vbv_occupancy	11	uimsbf
marker_bit	1	blsbf
latter_half_vbv_occupancy	15	uimsbf
marker_bit	1	blsbf
}		
}		
video_object_layer_shape	2	uimsbf
if (video_object_layer_shape == "grayscale" && video_object_layer_verid != ‘0001’)		
video_object_layer_shape_extension	4	uimsbf
marker_bit	1	bslbf
vop_time_increment_resolution	16	uimsbf
marker_bit	1	bslbf
fixed_vop_rate	1	bslbf
if (fixed_vop_rate)		
fixed_vop_time_increment	1-16	uimsbf
if (video_object_layer_shape != “binary only”) {		
if (video_object_layer_shape == “rectangular”) {		
marker_bit	1	bslbf
video_object_layer_width	13	uimsbf
marker_bit	1	bslbf
video_object_layer_height	13	uimsbf
marker_bit	1	bslbf
}		
interlaced	1	bslbf
obmc_disable	1	bslbf
if (video_object_layer_verid == ‘0001’)		
sprite_enable	1	bslbf
else		
sprite_enable	2	uimsbf
if (sprite_enable== “static” || sprite_enable == “GMC”) {		
if (sprite_enable != “GMC”) {		
sprite_width	13	uimsbf
marker_bit	1	bslbf
sprite_height	13	uimsbf
marker_bit	1	bslbf
sprite_left_coordinate	13	simsbf
marker_bit	1	bslbf
sprite_top_coordinate	13	simsbf
marker_bit	1	bslbf
}		
no_of_sprite_warping_points	6	uimsbf
sprite_warping_accuracy	2	uimsbf
sprite_brightness_change	1	bslbf

if (sprite_enable != “GMC”)		
low_latency_sprite_enable	1	bslbf
}		
if (video_object_layer_verid != ‘0001’ && video_object_layer_shape != ”rectangular”)		
sadct_disable	1	bslbf
not_8_bit	1	bslbf
if (not_8_ bit) {		
quant_precision	4	uimsbf
bits_per_pixel	4	uimsbf
}		
if (video_object_layer_shape==”grayscale”) {		
no_gray_quant_update	1	bslbf
composition_method	1	bslbf
linear_composition	1	bslbf
}		
quant_type	1	bslbf
if (quant_type) {		
load_intra_quant_mat	1	bslbf
if (load_intra_quant_mat)		
intra_quant_mat	8*[2-64]	uimsbf
load_nonintra_quant_mat	1	bslbf
if (load_nonintra_quant_mat)		
nonintra_quant_mat	8*[2-64]	uimsbf
if(video_object_layer_shape==”grayscale”) {		
for(i=0; i<aux_comp_count; i++) {		
load_intra_quant_mat_grayscale	1	bslbf
if(load_intra_quant_mat_grayscale)		
intra_quant_mat_grayscale[i]	8*[2-64]	uimsbf
load_nonintra_quant_mat_grayscale	1	bslbf
if(load_nonintra_quant_mat_grayscale)		
nonintra_quant_mat_grayscale[i]	8*[2-64]	uimsbf
}		
}		
}		
if (video_object_layer_verid != ‘0001’)		
quarter_sample	1	bslbf
complexity_estimation_disable	1	bslbf
if (!complexity_estimation_disable)		
define_vop_complexity_estimation_header()		
resync_marker_disable	1	bslbf
data_partitioned	1	bslbf
if(data_partitioned)		
reversible_vlc	1	bslbf
if(video_object_layer_verid != ’0001’) {		

newpred_enable	1	bslbf
if (newpred_enable) {		
requested_upstream_message_type	2	uimsbf
newpred_segment_type	1	bslbf
}		
reduced_resolution_vop_enable	1	bslbf
}		
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
}		
next_start_code()		
while (next_bits()== user_data_start_code){		
user_data()		
}		
if (sprite_enable == "static" && !low_latency_sprite_enable)		
VideoObjectPlane()		

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

"

VideoObjectLayer() {	**No. of bits**	**Mnemonic**
if(next_bits() == video_object_layer_start_code) {		
short_video_header = 0		
video_object_layer_start_code	32	bslbf
random_accessible_vol	1	bslbf
video_object_type_indication	8	uimsbf
if (video_object_type_indication == "Fine Granularity Scalable") {		
fgs_layer_type	2	uimsbf
video_object_layer_priority	3	uimsbf
aspect_ratio_info	4	uimsbf
if (aspect_ratio_info == "extended_PAR") {		
par_width	8	uimsbf
par_height	8	uimsbf
}		
vol_control_parameters	1	bslbf
if (vol_control_parameters) {		
chroma_format	2	uimsbf
low_delay	1	uimsbf
}		
marker_bit	1	bslbf
vop_time_increment_resolution	16	uimsbf
marker_bit	1	bslbf
fixed_vop_rate	1	bslbf
if (fixed_vop_rate)		
fixed_vop_time_increment	1-16	uimsbf
marker_bit	1	bslbf

video_object_layer_width	13	uimsbf
marker_bit	1	bslbf
video_object_layer_height	13	uimsbf
marker_bit	1	bslbf
interlaced	1	bslbf
if (fgs_layer_type ==“FGST” || fgs_layer_type ==“FGS_FGST”)		
fgs_ref_layer_id	4	uimsbf
if (fgs_layer_type ==“FGS” || fgs_layer_type ==“FGS_FGST”) {		
fgs_frequency_weighting_enable	1	bslbf
if (fgs_frequency_weighting_enable) {		
load_fgs_frequency_weighting_matrix	1	bslbf
if (load_fgs_frequency_weighting_matrix)		
fgs_frequency_weighting_matrix	3*[2-64]	uimsbf
}		
}		
if (fgs_layer_type ==“FGST” || fgs_layer_type ==“FGS_FGST”) {		
fgst_frequency_weighting_enable	1	bslbf
if (fgst_frequency_weighting_enable) {		
load_fgst_frequency_weighting_matrix	1	bslbf
if (load_fgst_frequency_weighting_matrix)		
fgst_frequency_weighting_matrix	3*[2-64]	uimsbf
}		
}		
quarter_sample	1	bslbf
fgs_resync_marker_disable	1	bslbf
do {		
if (nextbits_bytealigned() == group_of_vop_start_code)		
Group_of_VideoObjectPlane()		
FGSVideoObjectPlane()		
} while((nextbits_bytealigned()==group_of_vop_start_code)|| (nextbits_bytealigned()==fgs_vop_start_code))		
} else {		
is_object_layer_identifier	1	uimsbf
if (is_object_layer_identifier) {		
video_object_layer_verid	4	uimsbf
video_object_layer_priority	3	uimsbf
}		
aspect_ratio_info	4	uimsbf
if (aspect_ratio_info == “extended_PAR”) {		
par_width	8	uimsbf
par_height	8	uimsbf
}		

vol_control_parameters	1	bslbf
if (vol_control_parameters) {		
chroma_format	2	uimsbf
low_delay	1	uimsbf
vbv_parameters	1	blsbf
if (vbv_parameters) {		
first_half_bit_rate	15	uimsbf
marker_bit	1	bslbf
latter_half_bit_rate	15	uimsbf
marker_bit	1	bslbf
first_half_vbv_buffer_size	15	uimsbf
marker_bit	1	bslbf
latter_half_vbv_buffer_size	3	uimsbf
first_half_vbv_occupancy	11	uimsbf
marker_bit	1	blsbf
latter_half_vbv_occupancy	15	uimsbf
marker_bit	1	blsbf
}		
}		
video_object_layer_shape	2	uimsbf
if (video_object_layer_shape == "grayscale" && video_object_layer_verid != ‘0001’)		
video_object_layer_shape_extension	4	uimsbf
marker_bit	1	bslbf
vop_time_increment_resolution	16	uimsbf
marker_bit	1	bslbf
fixed_vop_rate	1	bslbf
if (fixed_vop_rate)		
fixed_vop_time_increment	1-16	uimsbf
if (video_object_layer_shape != “binary only”) {		
if (video_object_layer_shape == “rectangular”) {		
marker_bit	1	bslbf
video_object_layer_width	13	uimsbf
marker_bit	1	bslbf
video_object_layer_height	13	uimsbf
marker_bit	1	bslbf
}		
interlaced	1	bslbf
obmc_disable	1	bslbf
if (video_object_layer_verid == ‘0001’)		
sprite_enable	1	bslbf
else		
sprite_enable	2	uimsbf
if (sprite_enable== “static” || sprite_enable == “GMC”) {		
if (sprite_enable != “GMC”) {		
sprite_width	13	uimsbf
marker_bit	1	bslbf