INTERNATIONAL STANDARD

ISO/IEC 14496-2

Third edition
2004-06-01

AMENDMENT 1
2004-06-01

Information technology — Coding of audio-visual objects —

Part 2: **Visual**

AMENDMENT 1: Error resilient simple scalable profile

Technologies de l'information — Codage des objets audiovisuels —

Partie 2: Codage visuel

AMENDEMENT 1: Profil d'escalade simple résilient d'erreur

Reference number
ISO/IEC 14496-2:2004/Amd.1:2004(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.org
Web www.iso.org

Published 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 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 14496-2:2004 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Introduction

In ISO/IEC 14496-2:2004, the bitstream syntax of the Simple Scalable Profile does not allow the use of the error resilience tools. As the enhancement layer can only be decoded if it is received error-free this limits the use of scalable video to error-free communication environments. This would exclude mobile communications, a significant future market for MPEG-4 SSP.

Scalable video can be quite useful in many other applications. For instance, matching different or varying network bandwidths, video multicast to heterogeneous end systems and more importantly, providing different subjective quality of video content to subscribers of a given video service depending on their network tariffs and viewing preferences.

As such, ISO/IEC has defined the addition of an MPEG-4 Error Resilient Simple Scalable Profile. This Amendment describes the technical changes for this profile.

In addition, the new Level 0b is added to the MPEG-4 Simple Visual Profile and the amendments are described in this Amendment.

Information technology — Coding of audio-visual objects —

Part 2:
Visual

AMENDMENT 1: Error resilient simple scalable profile

In 6.2.5, replace:

if (video_object_layer_shape != “binary only”) {		
vop_quant	3-9	uimsbf
if(video_object_layer_shape==”grayscale”)		
for(i=0; i<aux_comp_count; i++)		
vop_alpha_quant[i]	6	uimsbf
if (vop_coding_type != “I”)		
vop_fcode_forward	3	uimsbf
if (vop_coding_type == “B”)		
vop_fcode_backward	3	uimsbf
if (!scalability) {		
if (video_object_layer_shape != “rectangular” && vop_coding_type != “I”)		
vop_shape_coding_type	1	bslbf
motion_shape_texture()		
while (nextbits_bytealigned() == resync_marker) {		
video_packet_header()		
motion_shape_texture()		
}		
}		
else {		
if (enhancement_type) {		
load_backward_shape	1	bslbf
if (load_backward_shape) {		
backward_shape_width	13	uimsbf
marker_bit	1	bslbf
backward_shape_ height	13	uimsbf
marker_bit	1	bslbf
backward_shape_horizontal_mc_spatial_ref	13	simsbf
marker_bit	1	bslbf
backward_shape_vertical_mc_spatial_ref	13	simsbf
backward_shape()		
load_forward_shape	1	bslbf
if (load_forward_shape) {		

forward_shape_width	13	uimsbf
marker_bit	1	bslbf
forward_shape_height	13	uimsbf
marker_bit	1	bslbf
forward_shape_horizontal_mc_spatial_ref	13	simsbf
marker_bit	1	bslbf
forward_shape_vertical_mc_spatial_ref	13	simsbf
forward_shape()		
}		
}		
}		
ref_select_code	2	uimsbf
combined_motion_shape_texture()		
}		
}		

with:

if (video_object_layer_shape != “binary only”) {		
vop_quant	3-9	uimsbf
if(video_object_layer_shape==”grayscale”)		
for(i=0; i<aux_comp_count; i++)		
vop_alpha_quant[i]	6	uimsbf
if (vop_coding_type != “I”)		
vop_fcode_forward	3	uimsbf
if (vop_coding_type == “B”)		
vop_fcode_backward	3	uimsbf
if (!scalability) {		
if (video_object_layer_shape != “rectangular” && vop_coding_type != “I”)		
vop_shape_coding_type	1	bslbf
}		
else {		
if (enhancement_type) {		
load_backward_shape	1	bslbf
if (load_backward_shape) {		
backward_shape_width	13	uimsbf
marker_bit	1	bslbf
backward_shape_ height	13	uimsbf
marker_bit	1	bslbf
backward_shape_horizontal_mc_spatial_ref	13	simsbf
marker_bit	1	bslbf
backward_shape_vertical_mc_spatial_ref	13	simsbf
backward_shape()		
load_forward_shape	1	bslbf
if (load_forward_shape) {		
forward_shape_width	13	uimsbf
marker_bit	1	bslbf

forward_shape_height	13	uimsbf
marker_bit	1	bslbf
forward_shape_horizontal_mc_spatial_ref	13	simsbf
marker_bit	1	bslbf
forward_shape_vertical_mc_spatial_ref	13	simsbf
forward_shape()		
}		
}		
}		
ref_select_code	2	uimsbf
}		
motion_shape_texture()		
while (nextbits_bytealigned() == resync_marker) {		
video_packet_header()		
motion_shape_texture()		
}		
}		

In 6.2.5.2, replace:

if (header_extension_code) {		
do {		
modulo_time_base	1	bslbf
} while (modulo_time_base != '0')		
marker_bit	1	bslbf
vop_time_increment	1-16	bslbf
marker_bit	1	bslbf
vop_coding_type	2	uimsbf
if (video_object_layer_shape != "rectangular") {		
change_conv_ratio_disable	1	bslbf
if (vop_coding_type != "I")		
vop_shape_coding_type	1	bslbf
}		
if (video_object_layer_shape != "binary only") {		
intra_dc_vlc_thr	3	uimsbf
if (sprite_enable == "GMC" && vop_coding_type == "S" && no_of_sprite_warping_points > 0)		
sprite_trajectory()		
if ((reduced_resolution_vop_enable) && (video_object_layer_shape == "rectangular") && ((vop_coding_type == "P") || (vop_coding_type == "I")))		
vop_reduced_resolution	1	bslbf
if (vop_coding_type != "I")		
vop_fcode_forward	3	uimsbf
if (vop_coding_type == "B")		
vop_fcode_backward	3	uimsbf
}		
}		

with:

if (header_extension_code) {		
do {		
modulo_time_base	1	bslbf
} while (modulo_time_base != '0')		
marker_bit	1	bslbf
vop_time_increment	1-16	bslbf
marker_bit	1	bslbf
vop_coding_type	2	uimsbf
if (video_object_layer_shape != "rectangular") {		
change_conv_ratio_disable	1	bslbf
if (vop_coding_type != "I")		
vop_shape_coding_type	1	bslbf
}		
if (video_object_layer_shape != "binary only") {		
intra_dc_vlc_thr	3	uimsbf
if (sprite_enable == "GMC" && vop_coding_type == "S" && no_of_sprite_warping_points > 0)		
sprite_trajectory()		
if ((reduced_resolution_vop_enable) && (video_object_layer_shape == "rectangular") && ((vop_coding_type == "P") || (vop_coding_type == "I")))		
vop_reduced_resolution	1	bslbf
if (vop_coding_type != "I")		
vop_fcode_forward	3	uimsbf
if (vop_coding_type == "B")		
vop_fcode_backward	3	uimsbf
}		
if (scalability)		
ref_select_code	2	uimsbf
}		

In 6.2.5.3, replace:

if (!not_coded) {		
if (sprite_enable == "GMC" && vop_coding_type == "S" && derived_mb_type < 2)		
mcsel	1	bslbf
if ((!(sprite_enable == "GMC" && vop_coding_type == "S" && mcsel) && derived_mb_type < 2) || derived_mb_type == 2)		
motion_coding("forward", derived_mb_type)		
}		
}		
} while (next_bits() != motion_marker)		

with:

if (!not_coded) {		
if (sprite_enable == “GMC” && vop_coding_type == “S” && derived_mb_type < 2)		
mcsel	1	bslbf
if ((!(sprite_enable == “GMC” && vop_coding_type == “S” && mcsel) && derived_mb_type < 2) || (derived_mb_type == 2) && !(scalability && ref_sel_code == 3))		
motion_coding(“forward”, derived_mb_type)		
}		
}		
} while (next_bits() != motion_marker)		

Replace Table 6-5 with the following table:

Visual_object_verid	Meaning
0000	reserved
0001	object type listed in Table 9-1
0010	object type listed in Table 9-2
0011	reserved
0100	object type listed in Table 9-3
0101	object type listed in Table 9-4
0110	object type listed in Table AMD1-1
0111 - 1111	reserved

Replace Table 6-11 with the following table:

Video Object Type	Code
Reserved	00000000
Simple Object Type	00000001
Simple Scalable Object Type	00000010
Core Object Type	00000011
Main Object Type	00000100
N-bit Object Type	00000101
Basic Anim. 2D Texture	00000110
Anim. 2D Mesh	00000111
Simple Face	00001000
Still Scalable Texture	00001001
Advanced Real Time Simple	00001010
Core Scalable	00001011
Advanced Coding Efficiency	00001100
Advanced Scalable Texture	00001101
Simple FBA	00001110

Simple Studio	00001111
Core Studio	00010000
Advanced Simple	00010001
Fine Granularity Scalable	00010010
Error resilient simple scalable	00010011
Reserved	00010100 - 11111111

Add the following table after Table 9-4:

Table AMD1-1 — Video Object Types

Visual Tools	Visual Object Types
	Error resilient simple scalable
I-VOP	X
P-VOP	X
B-VOP	X
DC Prediction	X
AC Prediction	X
4-MV, Unrestricted MV	X
Slice Resynchronization	X
Data Partitioning	X
Reversible VLC	X
Enhancement Layer Slice Resynchronization	X
Enhancement Layer Data Partitioning	X
Enhancement Layer Reversible VLC	X
Short Header	
Method 1/Method 2 Quantization	
Interlace	
Global Motion Compensation	
Quarter-pel Motion Compensation	
Fine Granularity Scalability	
FGS Temporal Scalability	

Add the following table after Table 9-8:

Table AMD1-2 — Definition of Error Resilient Simple Scalable Profile

ID	Object Types Profiles	Simple	Simple Scalable	Simple Scalable Error Resilience
SSER	Error resilient simple scalable	X	X	X

Add the following to Annex E:

E.4 Enhancement Layer Error Resilience

E.4.1 Enhancement Layer Slice Resynchronization

The Enhancement Layer Slice Resynchronization tool enables the usage of Slice Resynchronization, as defined in E.1.1, in enhancement layers. As the enhancement layer requires additional information, the HEC includes an additional 2 bits defining the ref_select_code.

The video packet resynchronization approach that can be applied using this tool is the same as the one described in E.1.1.

E.4.2 Enhancement Layer Data Partitioning

The Enhancement Layer Data Partitioning tool is the same as the Data Partitioning tool defined in E.1.2, with the exception that it now checks for the ref_select_code 3 special case in enhancement layer spatially scaled P-VOPs. There are no motion vectors in the VOP in this case.

E.4.3 Enhancement Layer Reversible VLC

The Enhancement Layer Reversible VLC tool is the same as the Reversible VLC tool defined in E.1.3.

In Table G-1, add the following:

Error Resilient Simple Scalable Profile/Level 0	00011101
Error Resilient Simple Scalable Profile/Level 1	00011110
Error Resilient Simple Scalable Profile/Level 2	00011111
Reserved	00100000

In Table G-1, replace:

Reserved	00010011 – 00100000

with:

Reserved	00010011 – 00011100

In Table G-1, replace:

Reserved	00001001 - 00001111

with:

Simple Profile/Level 0b	00001001
Reserved	00001010 - 00001111

In Table G-2, add the following row below the existing Error Resilience row:

Enhancement layer error resilience	ELER						×	✓	✓ d)	✓ e)	✓	×	✓	×	✓	×	×	✓	✓	✓	✓	✓	×	×	×	✓	×	✓ d)	×	×

In Table G-2, add the following column to the right of the existing Error Resilience column:

ELER
✓
✓
✓
✓
✓