INTERNATIONAL STANDARD ISO/IEC 14496-16:2006/Amd.1:2007
TECHNICAL CORRIGENDUM 1

Published 2008-06-15

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 16:
Animation Framework eXtension (AFX)

AMENDMENT 1: Geometry and shadow

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Codage des objets audiovisuels —

Partie 16: Extension du cadre d'animation (AFX)

AMENDEMENT 1: Géométrie et ombre

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 14496-16:2006/Amd.1:2007 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 14496-16:2006/Amd.1:2007/Cor.1:2008(E)**

Published in Switzerland

In 5.9.2.2.1.9, 3DMC_extension_header, replace:

3DMC_extension_header() {	No. of bits	Mnemonic
do {		
function_type	4	uimsbf
if (function_type == “Order_mode”)		
Order_mode_header ()		
else if (function_type == “Adaptive_quant_texCoord_mode”)		
Adaptive_quant_texCoord_mode_header()		
} while(function_type != “Escape_mode”)		
}		

with the following (where changes are highlighted in grey):

3DMC_extension_header() {	No. of bits	Mnemonic
do {		
function_type	4	vlclbf
if (function_type == “Order_mode”)		
Order_mode_header ()		
else if (function_type == “Adaptive_quant_texCoord_mode”)		
Adaptive_quant_texCoord_mode_header()		
else if (function_type == “Multiple_attribute_mode”)		
multiple_attribute_mode_header()		
} while(function_type != “Escape_mode”)		
}		

Insert the following new subclauses, renumbering subsequent subclauses:

5.9.2.2.1.11 multiple_attribute_mode_header

multiple_attribute_mode_header(){	No. of bits	Mnemonic
number_of_texCoord	5	
for 2 to number_of_texCoord		vlclbf
texCoord_header()		
number_of_otherAttr	8	
for 1 to number_of_otherAttr		
otherAttr_header()		
}		

5.9.2.2.1.12 otherAttr_header

otherAttr_header() {	No. of bits	Mnemonic
otherAttr_binding	2	uimsbf
otherAttr_dimension	8	
if (otherAttr_binding != "not_bound") {		
otherAttr_bbox	1	
if (otherAttr_bbox == '1') {		
for (i= &; i < otherAttr_dimension; i++)		
otherAttr_min(i)	32	bslbf
otherAttr_size	32	bslbf
}		
otherAttr_quant	5	uimsbf
otherAttr_pred_type	2	uimsbf
if (otherAttr_pred_type=="tree_prediction" || otherAttr_pred_type=="parallelogram_prediction") {		
otherAttr_nlambda	2	uimsbf
for (i=1; i<otherAttr_nlambda; i++)		
otherAttr_lambda	4-19	simsbf
}		
}		
}		

Replace 5.9.2.2.1.18, root_triangle():

root_triangle() {	No. of bits	Mnemonic
if (marching_triangle)		
qf_decode(**marching_pattern**, marching_pattern_context[**marching_pattern**])		vlclbf
else {		
if (3D_MOBL_start_code == "partition_type_2")		
if (tt_leaf == '0' && depth==0)		
qf_decode(**td_orientation**, td_orientation_context)		vlclbf
if (tt_leaf == '0')		
depth++		
else		
depth--		
}		
if (3D_MOBL_start_code == "partition_type_2")		
if (triangulated == '0')		
qf_decode(**polygon_edge**, polygon_edge_context[**polygon_edge**])		vlclbf
root_coord()		
root_normal()		
root_color()		
root_texCoord()		
}		

with the following (where changes are highlighted in grey):

root_triangle() {	No. of bits	Mnemonic
if (marching_triangle)		
qf_decode(**marching_pattern**, marching_pattern_context[**marching_pattern**])		vlclbf
else {		
if (3D_MOBL_start_code == "partition_type_2")		
if (tt_leaf == '0' && depth==0)		
qf_decode(**td_orientation**, td_orientation_context)		vlclbf
if (tt_leaf == '0')		
depth++		
else		
depth--		
}		
if (3D_MOBL_start_code == "partition_type_2")		
if (triangulated == '0')		
qf_decode(**polygon_edge**, polygon_edge_context[**polygon_edge**])		vlclbf
root_coord()		
root_normal()		
root_color()		
for (i =1; i<= number_of_texCoord; i++)		
root_texCoord(i)		
for (i = 1; i<= number_of_otherAttr; i++)		
root_otherAttr(i)		
}		

At the end of 5.9.2.2.1.18, add the following tables:

root_otherAttr() {	No. of bits
if (otherAttr_binding != "not_bound")	
if (3D_MOBL_start_code == "partition_type_2") {	
if (otherAttr_binding != "bound_per_vertex" || visited[vertex_index] == 0) {	
root_otherAttr_sample()	
if (otherAttr_binding!= "bound_per_vertex" || visited[vertex_index] == 0) {	
otherAttr_sample()	
otherAttr_sample()	
}	
}	
else {	
root_otherAttr_sample()	
otherAttr_sample()	
otherAttr_sample()	
}	
}	

'root_otherAttr_sample() {	No. of bits
for (i=0; i<otherAttr_order; i++)	
for (j=0; j<otherAttr_quant; j++)	
qf_decode(**otherAttr_bit**, zero_context)	
}	

otherAttr_sample() {	No. of bits
for (i=0; i< otherAttr_order; i++) {	
j=0	
do {	
qf_decode(**otherAttr_leading_bit**, otherAttr_leading_bit_context[2*j+i])	
j++	
} while (j<otherAttr_quant && otherAttr_leading_bit == '0')	
if (otherAttr_leading_bit == '1') {	
qf_decode(**otherAttr_sign_bit**, zero_context)	
do {	
qf_decode(**otherAttr_trailing_bit**, zero_context)	
} while (j<otherAttr_quant)	
}	
}	
}	

Replace 5.9.2.2.1.19, triangle():

triangle(i) {	No. of bits	Mnemonic
if (marching_triangle)		
qf_decode(**marching_edge**, marching_edge_context[**marching_edge**])		vlclbf
else {		
if (3D_MOBL_start_code == "partition_type_2")		
if (tt_leaf == '0' && depth==0)		
qf_decode(**td_orientation**, td_orientation_context)		vlclbf
if (tt_leaf == '0')		
depth++		
else		
depth--		
if (triangulated == '0')		
qf_decode(**polygon_edge**, polygon_edge_context[**polygon_edge**])		vlclbf
coord()		
normal()		
color()		
texCoord()		
}		

with the following (where changes are highlighted in grey):

triangle(i) {	No. of bits	Mnemonic
if (marching_triangle)		
qf_decode(**marching_edge**, marching_edge_context[**marching_edge**])		vlclbf
else {		
if (3D_MOBL_start_code == "partition_type_2")		
if (tt_leaf == '0' && depth==0)		
qf_decode(**td_orientation**, td_orientation_context)		vlclbf
if (tt_leaf == '0')		
depth++		
else		
depth--		
}		
if (triangulated == '0')		vlclbf
qf_decode(**polygon_edge**, polygon_edge_context[**polygon_edge**])		
coord()		
normal()		
color()		
for (i=1; i<= number_of_texCoord; i++)		
texCoord(i)		
for (i=1; i<= number_of_otherAttr; i++)		
otherAttr(i)		
}		

In 5.9.2.3.1.9 3, DMC_extension_header, replace:

function_type: This 4-bit unsigned integer indicates the function type supported in 3DMC extension. Table AMD1-17shows the admissible values for **function_type.**

Table AMD1-17 — Admissible values for function type

function_type_code	function_type
0000	Order_mode
0001	Adaptive_quant_texCoord_mode
0011~1110	Reserved
1111	Escape code

with the following (where changes are highlighted in grey)::

function_type: This 4-bit unsigned integer indicates the function type supported in 3DMC extension. Table AMD1-17 shows the admissible values for **function_type**