INTERNATIONAL STANDARD

ISO/IEC 14496-4

Second edition
2004-12-15

AMENDMENT 7
2005-04-15

Information technology — Coding of audio-visual objects —

Part 4: Conformance testing

AMENDMENT 7: AFX conformance extensions

Technologies de l'information — Codage des objets audiovisuels —

Partie 4: Essai de conformité

AMENDEMENT 7: Extensions de conformité AFX

Reference number
ISO/IEC 14496-4:2004/Amd.7:2005(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 7 to ISO/IEC 14496-4:2004 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 4:
Conformance testing

AMENDMENT 7: AFX conformance extensions

Add the following table at the end of Table 4 in subclause 4.4.3.1 and renumber the first column.

N°	Feature	Reference of Test sequence and associated method
1.	SFVec4f	Nurbs_curve_anim
2.	CoordinateInterpolator4D	Nurbs_curve_anim
3.	PositionInterpolator4D	Quadric_anim_st
4.	extendedUpdate : PROTOlistInsertion	proto_list_insertion_1, proto_list_insertion_2
5.	extendedUpdate : PROTOlistDeletion	proto_list_deletion_1, proto_list_deletion_2
6.	extendedUpdate : PROTODeletion	proto_deletion_1, proto_deletion_2
7.	extendedUpdate : MultipleFieldReplacement	multiple_replacement_1, multiple_replacement_2
8.	extendedUpdate : MultipleIndexedFieldReplacement	multiple_indexed_replacement_1, multiple_indexed_replacement_2
9.	extendedUpdate : GlobalQuantizationConfiguration	global_quant_1, global_quant_2
10.	extendedUpdate : NodeDeletionEx	node_deletion_ext_1, node_deletion_ext_2

Add the following table at the end of Table 6 in subclause 4.4.3.3.

Name	Provider	Content
Quadric_anim_st	SGDL / MINDEGO	Algebraic shape of the 2nd degree, through Quadric node, defined by 6 geometric control point in the projective space (4D). Deformation of the shape by moving the control points.
Nurbs_curve_anim	Mindego	A NURBS curve is animated by changing its control points over time.
proto_list_insertion_1	ENST	First sample of proto list insertion: simple
proto_list_insertion_2	ENST	Second sample of proto list insertion: complex
proto_list_deletion_1	ENST	First sample of proto list deletion: simple
proto_list_deletion_2	ENST	Second sample of proto list deletion: complex
proto_deletion_1	ENST	First sample of proto deletion: simple
proto_deletion_2	ENST	Second sample of proto deletion: complex
multiple_replacement_1	ENST	First sample of multiple replacement: simple
multiple_replacement_2	ENST	Second sample of multiple replacement: complex
multiple_indexed_replacement_1	ENST	First sample of multiple indexed replacement: simple
multiple_indexed_replacement_2	ENST	Second sample of multiple indexed replacement: complex
global_quant_1	ENST	First sample of global quantization configuration: adding global quantization
global_quant_2	ENST	Second sample of global quantization configuration: removing global quantization
node_deletion_ext_1	ENST	First sample of extended node deletion: simple
node_deletion_ext_2	ENST	Second sample of extended node deletion: complex

Add the following subclause after 4.12.2.

4.13 Conformance on SL Extensions

4.13.1 Extended SL - Synchronization Layer Conformance

4.13.1.1 Bitstream Conformance

4.13.1.1.1 Conformance Requirements

SL-packetized bitstreams shall comply with the specifications in subclause 7.3.2 of ISO/IEC 14496-1:2004.

4.13.1.1.2 Measurement procedure

Syntax of the SL Packets shall meet the requirements of subclause 7.3.2 of ISO/IEC 14496-1:2004.

4.13.1.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

4.13.1.2 Terminal Conformance

4.13.1.2.1 Conformance requirement

This subclause extends the conformance requirements set by the Synchronisation Layer conformance requirements. As such, the latter are implied here.

Each bitstream shall meet the syntactic and semantic requirements specified in ISO/IEC 14496-1:2004. The following subclauses describe a set of semantic tests to be performed on bitstreams.

Syntax of the BIFS stream shall meet the requirements in Clause 8 of ISO/IEC 14496-11:2004.

In the description of the semantic tests, it is assumed that the bitstreams contains no errors due to transmission or other causes. Note that packet losses can occur.

The SL layer shall recover the Access Units in the appropriate Decoding Buffer.

For each test the condition or conditions that must be satisfied are given, as well as the prerequisites or conditions in which the test can be applied.

Note that the application of these tests requires parsing of the bitstream at the appropriate levels, which in some cases may go as far as the slice level for video. In some cases of scrambled data, descrambling is required before performing the test. Parsing and interpretation of the configuration of the SL-packetized stream is also required.

4.13.1.2.2 Measurement procedure

4.13.1.2.2.1 Test of the coherence between ODProfile and SL extension.

When parsing the SLConfigDescriptor, an extension of SLConfigDescriptor is allowed only if `ODProfileLevelIndication` is different from 0x01.

Note that SLExtensionDescriptor is an abstract class specified so as to be the base class of sl extensions.

4.13.1.2.2.2 Coherence of the SLExtension.

Configuration	Conformance condition
DependencyPointer	
Mode==1	hasESID==0 In other words hasESID!=0 implies mode==0
hasESID==1	ESID is a valid Elementary Stream Identification. In other words an ESDescriptor with a corresponding ESID exists.
Mode==0 && hasESID==0	The ESDescriptor containing this `ExtendedSLConfigDescriptor` shall have a `streamDependenceFlag`==1. Additionally `dependsOn_ES_ID` should be a valide Elementary Stream Identification. In other words an ESDescriptor with a corresponding ESID exists. Moreover `dependencyLength` shall be greater than or equal to `timeStampLength` Refer to section 10.2.3 entitled "**SL Packet Header Configuration**"for the definition of the following terms : `streamDependenceFlag`, `dependsOn_ES_ID`, `timeStampLength`
MarkerDescriptor	
All configurations	`markerLength`==1

4.13.1.2.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

4.13.1.2.4 Syntax of the TRIF file.

Since M4Mux and Extension of SL require signalling of data not included in the MP4 file format. It is necessary to use the TRIF file format defined in ISO/IEC 14496-5:2001 (definition of TRIF file format).

The syntax of the TRIF file is the following :

File structure

`InitialObjectDescriptor`
`StreamMapTable`
`while (hasMorePacket)`
`{`
`  SLPacketHeader`
`  SLPayloadLength`
`  Payload`
`}`

`StreamMapTable`

`While (hasMoreESIDs)`
`{`
`  bit(8) ESID_HIBYTE`
`  bit(8) ESID_LOWBYTE`
`}`

4.13.2 M4Mux tool Conformance

4.13.2.1 Bitstream Conformance

4.13.2.1.1 Conformance Requirements

M4Mux-ed streams shall comply with the specifications in subclause 7.4.2 of ISO/IEC 14496-1:2004.

4.13.2.1.2 Measurement procedure

Syntax of the bitstream shall meet the requirements of subclause 7.4.2 of ISO/IEC 14496-1:2004.

4.13.2.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

4.13.2.2 Terminal Conformance

4.13.2.2.1 Conformance Requirements

Each bitstream shall meet the syntactic and semantic requirements specified in ISO/IEC 14496-1:2004. This subclause describes a set of semantic tests to be performed on bitstreams. To verify whether the syntax is correct is straight forward and therefore not required in this subclause. In the description of the semantic tests, it is assumed that the testbed bitstream contains no errors due to transmission or other causes.

The FlexDemux shall recover the SL Packets in the appropriate Decoding Buffer bit-exact as presented to the multiplex, and this for every Elementary Stream present in the M4Mux-ed stream under test.

For each test the condition or conditions that must be satisfied are given, as well as the prerequisites or conditions in which the test can be applied. Note that the application of these tests requires parsing of the bitstream at the appropriate levels, which in some cases may go as far as the slice level for video. In some cases of scrambled data, descrambling is required before performing the test. Parsing and interpretation of the configuration of the M4Mux stream is also required.

When a maximum bitrate is specified for an Elementary Stream, see subclause 7.2.6.5 of ISO/IEC 14496-1:2004. Conformant bitstreams shall obey this constraint.

4.13.2.2.2 Measurement procedure

4.13.2.2.2.1 Test of the M4Mux packet header:

A M4Mux stream is a succession of M4Mux packets. Each M4Mux packet is built from an **index** (the M4Mux Channel number) followed by the **length** of the M4Mux packet payload, followed by the M4Mux **payload** itself.

The **index** is on one byte.

The **length** field is on:

⇒ One byte, If

- there is no M4Mux ident descriptor,
- or if the M4Mux Ident descriptor indicates the use of the first M4Mux tool (type==0).

⇒ On one byte or on several bytes, If the M4Mux Ident descriptor indicates the use of the second M4Mux tool (type==1).

Index- test if the `index`

⇒ **is smaller than 238**, that it `corresponds to one M4Mux Channel declared in the M4Mux Channel descriptor.`

`length` If the length is

- non zero, `this the Simple mode.` Test if the length of the M4Mux packet, corresponds to the size of the M4Mux packet payload, i.e. if it addresses the beginning of the next M4Mux packet. In this Simple Mode this is equal to the length of the single encapsulated SL packet.
- zero, test if it is followed by one byte, where the five most significant bits are the `FMC_version_number`, and where the three least significant bits are equal to 1. Such `M4Mux packets with a length equal to zero can be duplicated, with identical FMC_version_number values.` If this `FMC_version_number` does not match the version of the referenced M4MuxChannelDescriptor that has most recently been received, the following M4Mux packets belonging to the same M4Mux Channel cannot be parsed. The test can either wait until the required `FMC_version_number` becomes available or discard the M4Mux packet.

⇒ **Is equal to 238**, which indicates a M4Mux packet with possible `fmxClockReference` samples and `fmxRate`, the M4Mux stream bitrate.

`length` Test if the first following field is an `fmxClockReference` sample, if the second following field is a `fmxRate field`, as they are defined in the M4Mux Timing descriptor. If the length is greater than the sum of the lengths of the `fmxClockReference` sample and of the `fmxRate` field defined in the M4Mux Timing descriptor, test if the remaining part of the M4Mux packet payload is composed of M4Mux decriptors (see the section about tests for M4Mux descriptors). Test if the length of the M4Mux packet, corresponds to the size of the M4Mux packet payload, i.e. if it addresses the beginning of the next M4Mux packet.

`fmxClockReference` – The sequence of `fmxClockReference` time stamps in a M4Mux stream constitutes a clock reference stream. In successive M4Mux packets of that clock reference stream, the `fmxClockReference field` contains encoded values which are samples of a system clock, the resolution of which is given by the FCRResolution field within the M4Mux Timing descriptor. The constraints on the accuracy of the successive `fmxClockReference samples`, allowing the reconstruction of the original system clock from the `fmxClockReference` samples are defined by the application.

fmxRate – test that the value encoded in the `fmxRate` field is sufficiently large that, if all bytes between this M4Mux packet and the next M4Mux packet of the clock reference stream are transmitted at that rate, they are delivered to the System Decoder Model before the time the first byte of the next M4Mux packet is delivered.

⇒ **Is equal to 239,** which indicates a M4Mux packet with stuffing.

length: Test if the length field adresses a number of stuffing bytes and the beginning of the next M4Mux packet.

⇒ **Is in the range of 240 to 255 (inclusive)**. This is the MuxCode Mode. Test if the `MuxCode` referenced as (`MuxCode = index` – 240) corresponds to one MuxCode declared in one MuxCodeTableEntry of the M4Mux Codetable descriptor.

length Test if it is followed by one byte, where the

- four most significant bits are the `version` field.
- four least significant bits are equal to 1.

Test if the length of the M4Mux packet, corresponds to the size of the M4Mux packet payload, i.e. if it addresses the beginning of the next M4Mux packet. This is equal to total length of the first byte (with the four least significant bits are equal to 1) plus the length of the multiple encapsulated SL packets.

version – If this version does not match the version of the referenced `MuxCodeTableEntry` that has most recently been received, the M4Mux packet payload cannot be parsed. The test can either wait until the required version of `MuxCodeTableEntry` becomes available or discard the M4Mux packet.

4.13.2.2.2.2 Test of the configuration of the M4Mux streams:

The global tests defined within the ISO/IEC 14496-1:2004 standard for the system descriptors, in terms of descriptor' tags and lengths, apply to each M4Mux descriptors.

Five different M4Mux descriptors define the possible configuration of a M4Mux stream:

⇒ The M4Mux Timing

⇒ The M4Mux Ident

⇒ The M4Mux Channel

⇒ the M4MuxCodetable

⇒ The M4Mux BufferSize

Such descriptors may be provided by out of band means or by in-band means.

♦ **tests for the M4MuxTimingDescriptor**

FCR_ES_ID: Test that there is one Elementary Stream with the same ES_ID declared as an OCR_ES_ID.
The test on the **FCRResolution**, **FCRLength** and **FmxRateLength** fields depend on the application.

♦ **tests for the M4MuxIDent descriptor**

The test on the MuxID field depends on the application.
Muxtype – the encoded value shall comply with the possible values defined in the Multiplexing type table defined within the ISO/IEC 14496 standard.

Muxmanagement – the encoded value shall comply with the possible values defined in the Multiplexing management mode table defined within the ISO/IEC 14496 standard.

♦ **Tests for the M4MuxChannel Descriptor**

Tests on the first byte:
Test that the values of the **version_number** field are incremented by one
Test that the validity period of each M4MuxChannel Descriptor identified by its **version_number** is defined:

- First by a 'non empty' M4MuxChannel Descriptor sent as a 'current' M4MuxChannel Descriptor
- Second by an 'empty' M4MuxChannel Descriptor sent as a 'current' M4MuxChannel Descriptor

Test if the two least significant bits are equal to 1.

Tests on the remaining bytes, when present:
Test that their number is a multiple of three bytes.

For each couple (ES_ID,M4MuxChannel):

- Test if the declared ES_ID corresponds to an existing ES_ID.
- Test that each ES_ID is only used once.
- Test that each M4MuxChannel is only used once.

♦ **Tests for the M4MuxCodeTable Descriptor**

Test that there is an integer number of **MuxCodeTableEntry**.
Test for all **MuxCodeTableEntry**, with an assigned version value, that there is only one **MuxCodeTableEntry** defined
Test that a **M4MuxChannel** is only used once in a Slot definition in each **MuxCodeTableEntry** and among the last versions of the different **MuxCodeTableEntry** defined.
Test for each **MuxCodeTableEntry**

- That there is an integer number of **substructures**.
- That there is an integer number of **slots.**

♦ **Tests for the M4MuxBufferSize Descriptor**

Test that there is an integer number of **M4MuxBufferDescriptor**.
Test that each **M4MuxChannel** referenced in a **M4MuxBufferDescriptor** is declared within the **M4MuxChannel Descriptor**
Test that a **M4MuxChannel** is only used once in a **M4MuxBufferDescriptor** definition.

4.13.2.2.3 Tolerance

There is no tolerance. The diagnosis is pass or fail.

4.13.3 Test Suites

4.13.3.1 SL Extension Feature list

The test suite shall verify the features in Table AMD 7-1

The following shall be tested:

- Presence in the bitstream.
- Appropriate value of the fields after decoding

Table AMD 7-1 – SL Extension Test Suite Information

N°	Feature	Reference of Test sequence and associated method
1.	SLExtensionDescriptor	SLExtension00
2.	DependencyPointer	SLExtension00
3.	MarkerDescriptor	SLExtension00
4.	ODProfileLevelIndication	SLExtension00

4.13.3.2 M4Mux Feature List

The following shall be tested:

- Presence in the bitstream.
- Appropriate value of the fields after decoding

Table AMD 7-2 – M4Mux Test Suite Information

N°	Feature	Reference of Test sequence and associated method
1.	MuxManagement	FMX_DYN dynamic management of M4Mux channel descriptors
2.	MuxType	FMX_DYN dynamic management of M4Mux channel descriptors
3.	MuxCodeTableEntry	FMX_DYN dynamic management of M4Mux channel descriptors

4.13.3.3 Bitstreams

Table AMD 7-3 – Bitstreams

Name	Provider	Content
FMX_DYN	France Telecom R&D	M4Mux file with dynamic management of M4Mux channel descriptors
SLExtension00	France Telecom R&D	SL extension file with DependencyPointer, Marker Descriptor

4.14 Conformance on Amendment to XMT

4.14.1 M4Mux descriptor tags in XMT

4.14.1.1 Conformance Requirements

The global tests defined within the ISO/IEC 14496-1:2004 standard for the system descriptors, in terms of descriptor' tags and lengths, apply to each M4Mux descriptors.

The five different M4Mux descriptors that define the possible configuration of a M4Mux stream:

⇒ The M4Mux Timing
⇒ The M4Mux Ident
⇒ The M4Mux Channel
⇒ The M4MuxCodetable
⇒ The M4Mux BufferSize

Can be sent and described using the XMT syntax.

4.14.1.2 Measurement procedure

4.14.1.2.1 tests for the M4MuxTimingDescriptor

FCR_ES_ID: Test that there is one Elementary Stream with the same ES_ID. Test that this Elementary stream is declared as an OCR_ES_ID.

4.14.1.2.2 tests for the M4MuxIDent descriptor

Further test may be conducted to confirm that the **Muxtype** and the **Muxmanagement** are in-line with associated M4Mux stream.

4.14.1.2.3 Tests for the M4MuxChannel Descriptor

4.14.1.2.3.1 Tests for each M4MuxChannel Descriptor

Tests on the first byte, that the validity period of each M4MuxChannel Descriptor identified by its **version_number** is correctly defined.

Tests on the remaining bytes, when present:

- Test if the declared ES_IDs correspond to an existing ES_ID.
- Test that each ES_ID is only used once.
- Test that each M4MuxChannel is only used once.

4.14.1.2.3.2 Tests for successive M4MuxChannel Descriptor

⇒ Test that the values of the **version_number** field are incremented by one at each descriptor

4.14.1.2.4 Tests for the M4MuxCodeTable Descriptor

⇒ Test for all **MuxCodeTableEntry**, with an assigned version value, that there is only one **MuxCodeTableEntry** defined

⇒ Test that a **M4MuxChannel** is only used once in a Slot definition in each **MuxCodeTableEntry** and among the last versions of the different **MuxCodeTableEntry** defined.

4.14.1.2.5 Tests for the M4MuxBufferSize Descriptor

⇒ Test that each **M4MuxChannel** referenced in a **M4MuxBufferDescriptor** is declared within the **M4MuxChannel Descriptor**

⇒ Test that a **M4MuxChannel** is only used once in a **M4MuxBufferDescriptor definition.**

4.14.1.3 Tolerance

There is no tolerance for syntax and semantic checking. The diagnosis is pass or fail.

Add the following subclause after 8.3.2.

8.4 AFX (Animation Framework eXtension)

8.4.1 Bitstream conformance

8.4.1.1 Conformance Requirements

BIFS streams shall comply with the specifications in Clause 8 of ISO/IEC 14496-11:2004 and Clause 4 of ISO/IEC 14496-16:2004.

8.4.1.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of Clause 8 of ISO/IEC 14496-11:2004 and Clause 4 of ISO/IEC 14496-16:2004.

8.4.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

8.4.2 Terminal conformance

8.4.2.1 Conformance Requirements

The terminal shall comply with the specifications in Clause 8 of ISO/IEC 14496-11:2004 and Clause 4 of ISO/IEC 14496-16:2004.

8.4.2.2 Measurement Procedure

The terminal shall decode successfully all the test suites listed below. A test suite is a suite of material and measurement algorithms and associated reference algorithms.

8.4.2.2.1 AFX Feature List

The test suite shall verify the features in Table AMD 7-4. For nodes, the following shall be tested:

- Presence in the scene tree after decoding.
- Appropriate value of the fields after decoding.

Table AMD 7-4 — AFX Test Suite Information

N°	Feature	Reference of Test sequence and associated method
1	BitWrapper	This node shall be tested together with the AFX bitstreams in subclause 8.5. Both url and buffer shall be tested for each bitstream.
2	DepthImage	This node shall be tested together with OctreeImage, PointTexture, and SimpleTexture nodes.
3	FFD	FFD
4	MeshGrid	Torus_C_LA_BIFS, Torus_C_OA_BIFS, Humanoid_LA_BIFS, Humanoid_OA_BIFS, Sphere_GA_BIFS, Quad_LA_BIFS, Quad_OA_BIFS, Quad_GA_BIFS, Cyclic_LA_BIFS, Cyclic_OA_BIFS, Cyclic_GA_BIFS
5	NonLinearDeformer	Bend, taper, twist, shell
6	NurbsCurve	NurbsCurve, NurbsCurve_anim
7	NurbsCurve2D	NurbsCurve2D, NurbsCurve2D_anim

8	NurbsSurface	NurbsSurface
9	OctreeImage	OI_BVO_Still OI_BVO_Anim OI_TBVO_Still OI_TBVO_Anim
10	PointTexture	DI_Ortho-PT_8, DI_Ortho-PT_32, DI_Persp–PT_8, DI_Persp–PT_32
11	PositionAnimator	PositionAnimator, PositionAnimator_discrete, PositionAnimator_linear, PositionAnimator_NURBS_interp, PositionAnimator_paced, PositionAnimator_spline
12	PositionAnimator2D	PositionAnimator2D, PositionAnimator2D _discrete, PositionAnimator2D _linear, PositionAnimator2D _NURBS_interp, PositionAnimator2D _paced, PositionAnimator2D _spline
13	ProceduralTexture	PT_Default, PT_Gradient1, PT_Gradient2, PT_Gradient3, PT_Gradient4, PT_Gradient5, PT_Horizon, PT_Marble, PT_PinkGranite, PT_Brickwork, PT_Fabric
14	SBBone	SkinnedModel
15	SBMuscle	SkinnedModel
16	SBSegment	SkinnedModel
17	SBSite	SkinnedModel
18	SBSkinnedModel	SkinnedModel
19	SBVCAnimation	SkinnedModel
20	ScalarAnimator	ScalarAnimator_discrete ScalarAnimator_linear ScalarAnimator_paced ScalarAnimator_spline
21	SimpleTexture	DI_Ortho-ST_Still DI_Ortho-ST_Anim DI_Persp-ST_Still DI_Persp–ST_Anim
22	SubdivisionSurface	Ss, SS_Goldfish, SS_Britney, SS_BritneyDance, SS_RooDance, SS_RooFlip, SS_Shark
23	SubdivSurfaceSector	ss_img, tagpipes, tagpipes_anim, icosa_normal, icosa_concave
24	WaveletSubdivisionSurface	This node shall be tested according to subclause 8.5.5.2

8.4.2.3 Bitstreams

Name	**Provider**	**Content**	**Original wrl file**
Bend	Mindego	NonLinearDeformer that bends a rectangular object	Bend.wrl
Cyclic_GA_BIFS	V.U.B.	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Animation of the gridCoord field.	Cyclic_MG, Cyclic_Lev0, Cyclic_Lev1, Cyclic_Lev2

Cyclic_LA_BIFS	V.U.B.	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field	Cyclic_MG, Cyclic_Lev0, Cyclic_Lev1, Cyclic_Lev2
Cyclic_OA_BIFS	V.U.B.	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Animation of the vertexOffset field.	Cyclic_MG, Cyclic_Lev0, Cyclic_Lev1, Cyclic_Lev2
DI_Ortho-PT_8	SAMSUNG AIT	Orthographic projection in DepthImage node. 8 bits representation of depth value in PointTexture node.	DI_Ortho-PT_8.wrl
DI_Ortho-PT_32	SAMSUNG AIT	Orthographic projection in DepthImage node. 32 bits representation of depth value in PointTexture node.	DI_Ortho-PT_32.wrl
DI_Ortho-ST_Still	SAMSUNG AIT	Orthographic projection in DepthImage node. Still version of SimpleTexture node.	DI_Ortho-ST_Still.wrl
DI_Ortho-ST_Anim	SAMSUNG AIT	Orthographic projection in DepthImage node. Animated version of SimpleTexture node.	DI_Ortho-ST_Anim.wrl
DI_Persp-PT_8	SAMSUNG AIT	Perspective projection in DepthImage node. 8 bits representation of depth value in PointTexture node.	DI_Persp-PT_8.wrl
DI_Persp-PT_32	SAMSUNG AIT	Perspective projection in DepthImage node. 32 bits representation of depth value in PointTexture node.	DI_Persp-PT_32.wrl
DI_Persp-ST_Still	SAMSUNG AIT	Perspective projection in DepthImage node. Still version of SimpleTexture node.	DI_Persp-ST_Still.wrl
DI_Persp-ST_Anim	SAMSUNG AIT	Perspective projection in DepthImage node. Animated version of SimpleTexture node.	DI_Persp-ST_Anim.wrl
Humanoid_LA_ BIFS	V.U.B.	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field	Humanoid_MG, Humanoid_Lev0, Humanoid_Lev1, Humanoid_Lev2
Humanoid_OA_ BIFS	V.U.B.	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid. Animation of the vertexOffset field.	Humanoid_MG, Humanoid_Lev0, Humanoid_Lev1, Humanoid_Lev2
Icosa_concave	Mindego	Tagpipes sample with theta of three sectors animated	Icosa_concave.wrl
Icosa_normal	Mindego	Tagpipes sample with normal of a sector animated	Icosa_normal.wrl
NurbsCurve	Mindego	Draw a NURBS curve	NurbsCurve.wrl
NurbsCurve_anim	Mindego	Animate a NURBS curve	NurbsCurve_anim.wrl
NurbsCurve2D	Mindego	Draw a 2D NURBS curve	NurbsCurve2D.wrl
NurbsCurve2D_ anim	Mindego	Animate a 2D NURBS curve	NurbsCurve2D_anim.wrl
NurbsSurface	Mindego	Draw a NURBS surface	NurbsSurface.wrl
OI_BVO_Still	SAMSUNG AIT	Non-use of BitWrapper node. Non-use of voxelImageIndex. Still version of OctreeImage node.	OI_BVO_Still.wrl
OI_BVO_Anim	SAMSUNG AIT	Non-use of BitWrapper node. Non-use of voxelImageIndex. Animated version of OctreeImage node.	OI_BVO_Anim.wrl
OI_TBVO_Still	SAMSUNG AIT	Non-use of BitWrapper node. Use of voxelImageIndex. Still version of OctreeImage node.	OI_TBVO_Still.wrl

OI_TBVO_Anim	SAMSUNG AIT	Non-use of BitWrapper node. Use of voxelImageIndex. Animated version of OctreeImage node.	OI_TBVO_Anim.wrl
PositionAnimator	Mindego	PositionAnimator used as an interpolator (keyType 0, keyValueType 0)	PositionAnimator.wrl
PositionAnimator_ discrete	Mindego	PositionAnimator with discrete timeline (keyType 1)	PositionAnimator_ discrete.wrl
PositionAnimator_ linear	Mindego	PositionAnimator with linear timeline (keyType 2)	PositionAnimator_ linear.wrl
PositionAnimator_ NURBS_interp	Mindego	PositionAnimator with NURBS path (keyValueType 1, 2, 3) and basic interpolator (keyType 0)	PositionAnimator_NURBS _interp.wrl
PositionAnimator_ paced	Mindego	PositionAnimator with paced animation (keyType 3) over a piecewise linear path (keyValueType 0)	PositionAnimator_paced. wrl
PositionAnimator_ spline	Mindego	PositionAnimator with a velocity spline (keyType 4) over a piecewise linear path (keyValueType 0)	PositionAnimator_spline. wrl
PT_Default	Superscape	Procedural texture with default values	PT_Default.wrl
PT_Gradient1	Superscape	Simple gradient - rectangle + single cell	PT_Gradient1.wrl
PT_Gradient2	Superscape	Simple gradient - brick + 16 cells + roughness + distortion	PT_Gradient2.wrl
PT_Gradient3	Superscape	Simple gradient - weave + 16 cells + roughness + distortion	PT_Gradient3.wrl
PT_Gradient4	Superscape	Simple gradient - hexagonal + 16 cells + roughness + distortion	PT_Gradient4.wrl
PT_Gradient5	Superscape	Simple gradient - ring + 4 cells + roughness	PT_Gradient5.wrl
PT_Horizon	Superscape	Horizon texture - high roughness + low/medium distortion	PT_Horizon.wrl
PT_Marble	Superscape	Marble texture - multiple bWeights + unequal warpmap knots	PT_Marble.wrl
PT_PinkGranite	Superscape	Granite texture - plasma based + high roughness	PT_PinkGranite.wrl
PT_Brickwork	Superscape	Brickwork texture - brick tiling + multiple aWeights + multiple knots + low roughness	PT_Brickwork.wrl
PT_Fabric	Superscape	Fabric texture - weave tiling + low roughness + low distortion + multiple aWeights	PT_Fabric.wrl
Quad_GA_BIFS	V.U.B.	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Animation of the gridCoord field.	Quad_MG, Quad_Lev0, Quad_Lev1, Quad_Lev2
Quad_LA_BIFS	V.U.B.	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field	Quad_MG, Quad_Lev0, Quad_Lev1, Quad_Lev2
Quad_OA_BIFS	V.U.B.	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Animation of the vertexOffset field.	Quad_MG, Quad_Lev0, Quad_Lev1, Quad_Lev2
Shell	Mindego	Combination of two NonLinearDeformer that twist and taper a rectangular object	Shell.wrl
SkinnedModel	INT	Skinned and articulated model defined by using the collection of SB nodes.	SkinnedModel.wrl
Sphere_GA_BIFS	V.U.B.	Multi-resolution non-homogeneous mesh. Animation of the gridCoord field.	Sphere_MG, Sphere_Lev0, Sphere_Lev1, Sphere_Lev2
Ss	Mindego	Shows a simple subdivision surface	Ss.wrl

SS_Britney	Superscape	Extended Loop subdivision static biped model	SS_Britney.wrl
SS_BritneyDance	Superscape	Extended Loop subdivision animated biped model	SS_BritneyDance.wrl
SS_Goldfish	Superscape	Extended Loop subdivision textured goldfish model	SS_Goldfish.wrl
Ss_img	Mindego	Shows a simple subdivision surface with 4 sectors with a texture mapped	Ss_img.wrl
SS_RooDance	Superscape	Extended Loop subdivision animated "kangaroo" model	SS_RooDance.wrl
SS_RooFlip	Superscape	Extended Loop subdivision animated "kangaroo"model	SS_RooFlip.wrl
SS_Shark	Superscape	Extended Loop subdivision animated shark model	SS_Shark.wrl
Tagpipes	Mindego	Two crossing cylinders as subdivision surfaces with sectors tagged	Tagpipes.wrl
Tagpipes_anim	Mindego	Same as tagpipes sample with flatness of sectors animated	Tagpipes_anim.wrl
Taper	Mindego	NonLinearDeformer that tapers a rectangular object	Taper.wrl
Torus_C_LA_BIFS	V.U.B.	Multi-resolution non-homogeneous mesh with uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field	Torus_MG, Torus_Lev0, Torus_Lev1, Torus_Lev2, Torus_Lev3, Torus_Lev4
Torus_C_OA_BIFS	V.U.B.	Multi-resolution non-homogeneous mesh with uniform distributed reference grid. Animation of the vertexOffset field.	Torus_MG, Torus_Lev0, Torus_Lev1, Torus_Lev2, Torus_Lev3, Torus_Lev4
Twist	Mindego	NonLinearDeformer that twists a rectangular object	Twist.wrl

8.4.2.4 Tolerance

There is no tolerance. The diagnosis is pass or fail.

8.4.3 Rendering conformance

8.4.3.1 Conformance Requirements

All tools with non-trivial algorithms shall be tested for rendering conformance.

8.4.3.2 Measurement procedure and tolerance

8.4.3.2.1 ProceduralTexture

Nodes : ProceduralTexture

Provider : Superscape

Bitstream : PT_Default, PT_Gradient1, PT_Gradient2, PT_Gradient3, PT_Gradient4, PT_Gradient5, PT_Horizon, PT_Marble, PT_PinkGranite, PT_Brickwork, PT_Fabric

Procedural textures are a function of the supplied parameters.

While rendering is not required to be color exact - color depth will vary between terminals - the texture structure must be pixel exact. In particular the (distorted) cell outline, cell positioning, and the interior of plasma based textures must be preserved.

8.5 AFX Bitstreams

8.5.1 Common Conformance Point

8.5.1.1 Bitstream conformance

8.5.1.1.1 Conformance Requirements

AFX bitstreams shall comply with the objectTypeIndication and DecoderSpecificInfo specification in subclause 7.2.6 of ISO/IEC 14496-1:2004.

8.5.1.1.2 Measurement procedure

Syntax of the AFX bitstreams shall meet the requirements of subclause 7.2.6 of ISO/IEC 14496-1:2004.

8.5.1.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

8.5.2 Interpolator Compression

Interpolator represents the key frame-based animation that is the most widely used method in graphics animation and can be represented as various types of Interpolator nodes in BIFS. The data are represented as a piecewise linear animation path through motion capture or other authoring tools. It has key frames that consist of key and key value pair, where the key is a time stamp and the key value is the corresponding value to the key. Depending on the type of animation, the key values may represent various data types: scalar values, 3D coordinates, colors, normal vectors, or orientation (or rotation) values.

The main purpose of the Interpolator Compression(IC) tool is to provide efficient data transmission for key-frame based animation stream. More specifically, IC provides efficient data transmission for three types of interpolators – coordinate interpolator, orientation interpolator, and position interpolator.

Basically, IC involves one or two distinct functionalities that have emerged from the Core Experiment process for inclusion in ISO/IEC 14496-11:2004. The functionalities are:

1. Key preserving ; coordinate interpolator, orientation interpolator, position interpolator

    A. Compress interpolators while preserve key and keyValue structure for random access.

2. Path preserving ; orientation interpolator, position interpolator

    A. Compress interpolators without necessarily preserving all keys and key values for maintaining animation path.

8.5.2.1 Conformance Points

8.5.2.1.1 Covered Functionalities

The conformance points for IC covers the key preserving and path preserving. These functionalities relate to the compressed representation of interpolator node carried by BitWrapper node as described in ISO/IEC 14496-11:2004.

As for carriage of compressed representation of interpolator node using BitWrapper node, it can be carried either in a separate stream or within the scene stream (BIFS stream). Therefore, IC also shall be tested together with this node as described in ISO/IEC 14496-11:2004.

The following subclauses specify the normative tests for verifying conformance of IC bitstreams and IC decoders. Those normative tests make use of test data (bitstream test suites) provided as an electronic annex to this document, and of a software verifier specified in ISO/IEC 14496-5:2001/Amd.7 with source code available in electronic format.

8.5.2.2 Bitstream conformance

8.5.2.2.1 Conformance Requirements

BIFS streams shall comply with the specifications in 8.9 and 7.2.2.20 of ISO/IEC 14496-11:2004.

8.5.2.2.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of 8.9 and 7.2.2.20 of ISO/IEC 14496-11:2004.

8.5.2.2.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

8.5.2.3 Terminal conformance

8.5.2.3.1 Coordinate Interpolator

8.5.2.3.1.1 Conformance Requirements

Since the functionalities are not directly supported by a profile@level combination, the conformance of the corresponding decoder shall be defined, such that the decoder conforms to the functionality. A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-11:2004 and meets all the general requirements, defined in that document, which apply for the functionality considered, and if it can decode bitstreams with any options or parameters with values permitted for that functionality. In the case of using BIFS for scene representation, the decoding process that is specified in Clause 8 of ISO/IEC 14496-1:2004 shall also be implemented.

In the following subclauses the term ‘reference decoder’ means the technical report software verifier (ISO/IEC 14496-5:2001/Amd.7). The reference decoder is a decoder that implements precisely the decoding process as specified in ISO/IEC 14496-11:2004.

8.5.2.3.1.2 Measurement Procedure

To test the validity of the output of a coordinate interpolator decoder that is not included in a profile@level combination, the conformance will be performed only for the functionality independent from the existing profiles. Every decoder shall be able to decode the bitstreams provided for key preserving functionality as described in subclause 8.5.2.3.1.3.

8.5.2.3.1.3 Test Bitstreams for key preserving

♦ **Test Bitstreams keypreserving#1**

Purpose: Exercise the key preserving functionality of MPEG-4 Coordinate Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
ci_kp#1-1	key quantization bit size decoding	nKeyQBit is set to 10	ci_kp#1-nKQBit	ci_kp#1-nKQBit
ci_kp#1-2	linear key decoding	bIsLinearSubRegion is enabled	ci_kp#1-LinearKey	ci_kp#1-LinearKey
ci_kp#1-3	linear key decoding	bIsLinearSubRegion is disabled	ci_kp#1-NoLinearKey	ci_kp#1-NoLinearKey
ci_kp#1-4	key range decoding	bRangeFlag is enabled	ci_kp#1-OverKeyRange	ci_kp#1-OverKeyRange
ci_kp#1-5	key range decoding	bRangeFlag is disabled	ci_kp#1-KeyRange	ci_kp#1-KeyRange
ci_kp#1-6	n-th order DPCM	nDPCMOrder is 2nd order	ci_kp#1-2ndDPCM	ci_kp#1-2ndDPCM
ci_kp#1-7	shift operation	bShiftFlag is enabled	ci_kp#1-Shift	ci_kp#1-Shift
ci_kp#1-8	shift operation	bShiftFlag is disabled	ci_kp#1-NoShift	ci_kp#1-NoShift
ci_kp#1-9	if DND and folding operations are not selected	nDNDOrder is -1	ci_kp#1-MinusOneDNDOrder	ci_kp#1-MinusOneDNDOrder
ci_kp#1-10	folding operation	nDNDOrder is 0	ci_kp#1-ZeroDNDOrder	ci_kp#1-ZeroDNDOrder
ci_kp#1-11	DND operation after folding operation	nDNDOrder is 3	ci_kp#1-PositiveDNDOrder	ci_kp#1-PositiveDNDOrder
ci_kp#1-12	key Invert down operation	bKeyInvertDownFlag is enabled	ci_kp#1-KeyInvertDown	ci_kp#1-KeyInvertDown
ci_kp#1-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	ci_kp#1-NoKeyInvertDownAndSignedAAC	ci_kp#1-NoKeyInvertDownAndSignedAAC
ci_kp#1-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	ci_kp#1-NoKeyInvertDownAndUnsignedAAC	ci_kp#1-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams keypreserving#2**

Purpose: Exercise the key preserving functionality of MPEG-4 Coordinate Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
ci_kp#2-1	transpose mode decoding	bTranspose is enabled	ci_kp#2-bTranspose	ci_kp#2-bTranspose
ci_kp#2-2	vertex mode decoding	bTranspose is disabled	ci_kp#2-bVertex	ci_kp#2-bVertex
ci_kp#2-3	kev value quantization bit size decoding	nKVQBit is set to 10	ci_kp#2-nKVQBit	ci_kp#2-nKVQBit
ci_kp#2-4	floating point number decoding, which consists of mantissa and exponent.	bUse32Float is enabled	ci_kp#2-FPNcoding	ci_kp#2-FPNcoding
ci_kp#2-5	the decoding of the digits of the minimum values of each component, when all these values have the different digit	bAllSameMantissaDigitFlag is disabled	ci_kp#2-allsamemantissadigit	ci_kp#2-allsamemantissadigit
ci_kp#2-6	the decoding of the digits of the minimum values of each component, when all these values have the same digit	bSameKVDigitFlag is disabled	ci_kp#2-samekvdigit	ci_kp#2-samekvdigit
ci_kp#2-7	the decoding of the digit of the maximum range among the ranges of each components	bMaxDigitFlag is disabled	ci_kp#2-maxdigit	ci_kp#2-maxdigit
ci_kp#2-8	the sign of exponent of all the minimum values and the maximum range	bAllSameExponentSign is disabled	ci_kp#2-allsameexponentsign	ci_kp#2-allsameexponentsign
ci_kp#2-9	the decoding of all the minimum values and the maximum range	nDigit is set to 6	ci_kp#2-digit	ci_kp#2-digit
ci_kp#2-10	Test selection flag when some quantization values are same for each vertex and each component	bSelFlag ’s are some 0’s	ci_kp#2-bSelFlag	ci_kp#2-bSelFlag
ci_kp#2-11	selection flag when all quantization values are not same for each vertex and each component	bSelFlag ‘s are all 1’s	ci_kp#2-NotSamebSelFlag	ci_kp#2-NotSamebSelFlag
ci_kp#2-12	temporal DPCM mode	nDPCMMode ‘s are all 1’s	ci_kp#2-TMode	ci_kp#2-TMode
ci_kp#2-13	spatial DPCM mode	nDPCMMode ‘s are all 2’s	ci_kp#2-SMode	ci_kp#2-SMode
ci_kp#2-14	spatio-temporal DPCM mode	nDPCMMode ‘s are all 3’s	ci_kp#2-TSMode	ci_kp#2-TSMode

ci_kp#2-15	incremental mode decoding	nDicModeSelect ‘ s are all 1’s	ci_kp#2-IncrementalMode	ci_kp#2-IncrementalMode
ci_kp#2-16	occurrence mode decoding	nDicModeSelect ‘ s are all 0’s	ci_kp#2-OccurrenceMode	ci_kp#2-OccurrenceMode
ci_kp#2-17	key value coding bit when some differentiated key values are zeros after DPCM operation	nKVACodingBit ‘s are some 0’s	ci_kp#2-ZeroKVACBit	ci_kp#2-ZeroKVACBit
ci_kp#2-18	key value coding bit when all differentiated key values are not zeros after DPCM operation	nKVACodingBit ‘s are all positive integers	ci_kp#2-PositiveKVACBit	ci_kp#2-PositiveKVACBit
ci_kp#2-19	dictionary symbol decoding whether each dictionary symbol is referenced only once for each vertex and each component or not	bSoleKV ‘s are some 1’s	ci_kp#2-bSoleKV	ci_kp#2-bSoleKV
ci_kp#2-20	All dictionary symbols are not same for each vertex and each component	bSoleKV ‘s are all 1’s	ci_kp#2-AllSamebSoleKV	ci_kp#2-AllSamebSoleKV
ci_kp#2-21	if the reversion of symbol(between 0 and 1) is happened in the dictionary position indices	nTrueOne ‘s are all 0’s	ci_kp#2-ZeroTrueOne	ci_kp#2-ZeroTrueOne

♦ **Test Bitstreams keypreserving#3**

Purpose: Exercise the key preserving functionality of MPEG-4 Coordinate Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
ci_kp#3-1	key quantization bit size decoding	nKeyQBit is set to 10	ci_kp#3-nKQBit	ci_kp#3-nKQBit
ci_kp#3-2	linear key decoding	bIsLinearSubRegion is enabled	ci_kp#3-LinearKey	ci_kp#3-LinearKey
ci_kp#3-3	linear key decoding	bIsLinearSubRegion is disabled	ci_kp#3-NoLinearKey	ci_kp#3-NoLinearKey
ci_kp#3-4	key range decoding	bRangeFlag is enabled	ci_kp#3-OverKeyRange	ci_kp#3-OverKeyRange
ci_kp#3-5	key range decoding	bRangeFlag is disabled	ci_kp#3-KeyRange	ci_kp#3-KeyRange
ci_kp#3-6	n-th order DPCM	nDPCMOrder is 2nd order	ci_kp#3-2ndDPCM	ci_kp#3-2ndDPCM
ci_kp#3-7	shift operation	bShiftFlag is enabled	ci_kp#3-Shift	ci_kp#3-Shift
ci_kp#3-8	shift operation	bShiftFlag is disabled	ci_kp#3-NoShift	ci_kp#3-NoShift

ci_kp#3-9	if DND and folding operations are not selected	nDNDOrder is -1	ci_kp#3-MinusOneDNDOrder	ci_kp#3-MinusOneDNDOrder
ci_kp#3-10	folding operation	nDNDOrder is 0	ci_kp#3-ZeroDNDOrder	ci_kp#3-ZeroDNDOrder
ci_kp#3-11	DND operation after folding operation	nDNDOrder is 3	ci_kp#3-PositiveDNDOrder	ci_kp#3-PositiveDNDOrder
ci_kp#3-12	key Invert down operation	bKeyInvertDownFlag is enabled	ci_kp#3-KeyInvertDown	ci_kp#3-KeyInvertDown
ci_kp#3-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	ci_kp#3-NoKeyInvertDownAndSignedAAC	ci_kp#3-NoKeyInvertDownAndSignedAAC
ci_kp#3-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	ci_kp#3-NoKeyInvertDownAndUnsignedAAC	ci_kp#3-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams keypreserving#4**

Purpose: Exercise the key preserving functionality of MPEG-4 Coordinate Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
ci_kp#4-1	transpose mode decoding	bTranspose is enabled	ci_kp#4-bTranspose	ci_kp#4-bTranspose
ci_kp#4-2	vertex mode decoding	bTranspose is disabled	ci_kp#4-bVertex	ci_kp#4-bVertex
ci_kp#4-3	kev value quantization bit size decoding	nKVQBit is set to 10	ci_kp#4-nKVQBit	ci_kp#4-nKVQBit
ci_kp#4-4	floating point number decoding, which consists of mantissa and exponent.	bUse32Float is enabled	ci_kp#4-FPNcoding	ci_kp#4-FPNcoding
ci_kp#4-5	the decoding of the digits of the minimum values of each component, when all these values have the different digit	bAllSameMantissaDigitFlag is disabled	ci_kp#4-allsamemantissadigit	ci_kp#4-allsamemantissadigit
ci_kp#4-6	the decoding of the digits of the minimum values of each component, when all these values have the same digit	bSameKVDigitFlag is disabled	ci_kp#4-samekvdigit	ci_kp#4-samekvdigit
ci_kp#4-7	the decoding of the digit of the maximum range among the ranges of each components	bMaxDigitFlag is disabled	ci_kp#4-maxdigit	ci_kp#4-maxdigit

ci_kp#4-8	the sign of exponent of all the minimum values and the maximum range	bAllSameExponentSign is disabled	ci_kp#4-allsameexponentsign	ci_kp#4-allsameexponentsign
ci_kp#4-9	the decoding of all the minimum values and the maximum range	nDigit is set to 6	ci_kp#4-digit	ci_kp#4-digit
ci_kp#4-10	selection flag when some quantization values are same for each vertex and each component	bSelFlag 's are – some 0's	ci_kp#4-bSelFlag	ci_kp#4-bSelFlag
ci_kp#4-11	selection flag when all quantization values are not same for each vertex and each component	bSelFlag 's are all 1's	ci_kp#4-NotSamebSelFlag	ci_kp#4-NotSamebSelFlag
ci_kp#4-12	temporal DPCM mode	nDPCMMode 's are all 1's	ci_kp#4-TMode	ci_kp#4-TMode
ci_kp#4-13	spatial DPCM mode	nDPCMMode 's are all 2's	ci_kp#4-SMode	ci_kp#4-SMode
ci_kp#4-14	spatio-temporal DPCM mode	nDPCMMode 's are all 3's	ci_kp#4-TSMode	ci_kp#4-TSMode
ci_kp#4-15	incremental mode decoding	nDicModeSelect 's are all 1's	ci_kp#4-IncrementalMode	ci_kp#4-IncrementalMode
ci_kp#4-16	occurrence mode decoding	nDicModeSelect 's are all 0's	ci_kp#4-OccurrenceMode	ci_kp#4-OccurrenceMode
ci_kp#4-17	key value coding bit when some differentiated key values are zeros after DPCM operation	nKVACodingBit 's are some 0's	ci_kp#4-ZeroKVACBit	ci_kp#4-ZeroKVACBit
ci_kp#4-18	key value coding bit when all differentiated key values are not zeros after DPCM operation	nKVACodingBit 's are all positive integers	ci_kp#4-PositiveKVACBit	ci_kp#4-PositiveKVACBit
ci_kp#4-19	dictionary symbol decoding whether each dictionary symbol is referenced only once for each vertex and each component or not	bSoleKV 's are some 1's.	ci_kp#4-bSoleKV	ci_kp#4-bSoleKV
ci_kp#4-20	All dictionary symbols are not same for each vertex and each component	bSoleKV 's are all 1's.	ci_kp#4-AllSamebSoleKV	ci_kp#4-AllSamebSoleKV
ci_kp#4-21	if the reversion of symbol(between 0 and 1) is happened in the dictionary position indices	nTrueOne 's are all 0's	ci_kp#4-ZeroTrueOne	ci_kp#4-ZeroTrueOne

8.5.2.3.1.4 Tolerance

The diagnosis is to check whether the field data (key, keyValue) of CoordinateInterpolator node that is decoded from ".mp4" files correspond with the node included in the provided reference file.

8.5.2.3.2 Orientation Interpolator

8.5.2.3.2.1 Conformance Requirements

Since the functionalities are not directly supported by a profile@level combination, the conformance of the corresponding decoder shall be defined, such that the decoder conforms to the functionality.

A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-11:2004 and meets all the general requirements, defined in that document, which apply for the functionality considered, and if it can decode bitstreams with any options or parameters with values permitted for that functionality. In the case of using BIFS for scene representation, the decoding process that is specified in Clause 8 of ISO/IEC 14496-1:2004 shall also be implemented.

In the following subclauses the term 'reference decoder' means the technical report software verifier (ISO/IEC 14496-5:2001/Amd.7). The reference decoder is a decoder that implements precisely the decoding process as specified in ISO/IEC 14496-11:2004.

8.5.2.3.2.2 Measurement Procedure

To test the validity of the output of a orientation interpolator decoder which is not included in a profile@level combination, the conformance will be performed only for the functionality independent from the existing profiles. Every decoder shall be able to decode the bitstreams provided for key preserving and path-preserving functionalities as described in subclause 8.5.2.3.2.3 and subclause 8.5.2.3.2.4.

8.5.2.3.2.3 Test Bitstreams for key preserving

- **Test Bitstreams keypreserving#1**

Purpose: Exercise the key preserving functionality of MPEG-4 Orientation Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
oi_kp#1-1	key quantization bit size decoding	nKeyQBit is set to 10	oi_kp#1-nKQBit	oi_kp#1-nKQBit
oi_kp#1-2	linear key decoding	bIsLinearSubRegion is enabled	oi_kp#1-LinearKey	oi_kp#1-LinearKey
oi_kp#1-3	linear key decoding	bIsLinearSubRegion is disabled	oi_kp#1-NoLinearKey	oi_kp#1-NoLinearKey
oi_kp#1-4	key range decoding	bRangeFlag is enabled	oi_kp#1-OverKeyRange	oi_kp#1-OverKeyRange
oi_kp#1-5	key range decoding	bRangeFlag is disabled	oi_kp#1-KeyRange	oi_kp#1-KeyRange
oi_kp#1-6	n-th order DPCM	nDPCMOrder is 2nd order	oi_kp#1-2ndDPCM	oi_kp#1-2ndDPCM
oi_kp#1-7	shift operation	bShiftFlag is enabled	oi_kp#1-Shift	oi_kp#1-Shift
oi_kp#1-8	shift operation	bShiftFlag is disabled	oi_kp#1-NoShift	oi_kp#1-NoShift

oi_kp#1-9	if DND and folding operations are not selected	nDNDOrder is -1	oi_kp#1-MinusOneDNDOrder	oi_kp#1-MinusOneDNDOrder
oi_kp#1-10	folding operation	nDNDOrder is 0	oi_kp#1-ZeroDNDOrder	oi_kp#1-ZeroDNDOrder
oi_kp#1-11	DND operation after folding operation	nDNDOrder is 3	oi_kp#1-PositiveDNDOrder	oi_kp#1-PositiveDNDOrder
oi_kp#1-12	key Invert down operation	bKeyInvertDownFlag is enabled	oi_kp#1-KeyInvertDown	oi_kp#1-KeyInvertDown
oi_kp#1-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	oi_kp#1-NoKeyInvertDownAndSignedAAC	oi_kp#1-NoKeyInvertDownAndSignedAAC
oi_kp#1-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	oi_kp#1-NoKeyInvertDownAndUnsignedAAC	oi_kp#1-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams keypreserving#2**

Purpose: Exercise the key preserving functionality of MPEG-4 Orientation Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
oi_kp#2-1	key value quantization bit size decoding	nKVQBit is set to 10	oi_kp#2-nKVQBit	oi_kp#2-nKVQBit
oi_kp#2-2	1st-order DPCM	nKVDPCMOrder is set to 0	oi_kp#2-1stDPCM	oi_kp#2-1stDPCM
oi_kp#2-3	2nd-order DPCM (with Circular DPCM)	nKVDPCMOrder is set to 1	oi_kp#2-2ndDPCM	oi_kp#2-2ndDPCM
oi_kp#2-4	key value flag, indicating whether all quantized values are same in the each component of key values	x_keyvalue_flag is enabled y_keyvalue_flag is disabled z_keyvalue_flag is enabled	oi_kp#2-allsame	oi_kp#2-allsame
oi_kp#2-5	Binary AAC decoding	bIsUnaryAACs are disabled for all components	oi_kp#2-binaryaac	oi_kp#2-binaryaac
oi_kp#2-6	Unary AAC decoding	bIsUnaryAACs are enabled for all components	oi_kp#2-unaryaac	oi_kp#2-unaryaac
oi_kp#2-7	the flag, indicating whether the number of key values is more than 2	bIsMoreTwoKVs is enabled	oi_kp#2-bIsMoreTwoKVs	oi_kp#2-bIsMoreTwoKVs

♦ **Test Bitstreams keypreserving#3**

Purpose: Exercise the key preserving functionality of MPEG-4 Orientation Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
oi_kp#3-1	key quantization bit size decoding	nKeyQBit is set to 10	oi_kp#3-nKQBit	oi_kp#3-nKQBit
oi_kp#3-2	linear key decoding	bIsLinearSubRegion is enabled	oi_kp#3-LinearKey	oi_kp#3-LinearKey
oi_kp#3-3	linear key decoding	bIsLinearSubRegion is disabled	oi_kp#3-NoLinearKey	oi_kp#3-NoLinearKey
oi_kp#3-4	key range decoding	bRangeFlag is enabled	oi_kp#3-OverKeyRange	oi_kp#3-OverKeyRange
oi_kp#3-5	key range decoding	bRangeFlag is disabled	oi_kp#3-KeyRange	oi_kp#3-KeyRange
oi_kp#3-6	n-th order DPCM	nDPCMOrder is 2nd order	oi_kp#3-2ndDPCM	oi_kp#3-2ndDPCM
oi_kp#3-7	shift operation	bShiftFlag is enabled	oi_kp#3-Shift	oi_kp#3-Shift
oi_kp#3-8	shift operation	bShiftFlag is disabled	oi_kp#3-NoShift	oi_kp#3-NoShift
oi_kp#3-9	if DND and folding operations are not selected	nDNDOrder is -1	oi_kp#3-MinusOneDNDOrder	oi_kp#3-MinusOneDNDOrder
oi_kp#3-10	folding operation	nDNDOrder is 0	oi_kp#3-ZeroDNDOrder	oi_kp#3-ZeroDNDOrder
oi_kp#3-11	DND operation after folding operation	nDNDOrder is 3	oi_kp#3-PositiveDNDOrder	oi_kp#3-PositiveDNDOrder
oi_kp#3-12	key Invert down operation	bKeyInvertDownFlag is enabled	oi_kp#3-KeyInvertDown	oi_kp#3-KeyInvertDown
oi_kp#3-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	oi_kp#3-NoKeyInvertDownAndSignedAAC	oi_kp#3-NoKeyInvertDownAndSignedAAC
oi_kp#3-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	oi_kp#3-NoKeyInvertDownAndUnsignedAAC	oi_kp#3-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams keypreserving#4**

Purpose: Exercise the key preserving functionality of MPEG-4 Orientation Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
oi_kp#4-1	key value quantization bit size decoding	nKVQBit is set to 10	oi_kp#4-nKVQBit	oi_kp#4-nKVQBit
oi_kp#4-2	1st-order DPCM	nKVDPCMOrder is set to 0	oi_kp#4-1stDPCM	oi_kp#4-1stDPCM
oi_kp#4-3	2nd-order DPCM (with Circular DPCM)	nKVDPCMOrder is set to 1	oi_kp#4-2ndDPCM	oi_kp#4-2ndDPCM
oi_kp#4-4	key value flag, indicating whether all quantized values are same in the each component of key values	x_keyvalue_flag is enabled y_keyvalue_flag is disabled z_keyvalue_flag is enabled	oi_kp#4-allsame	oi_kp#4-allsame
oi_kp#4-5	Binary AAC decoding	bIsUnaryAACs are disabled for all components	oi_kp#4-binaryaac	oi_kp#4-binaryaac
oi_kp#4-6	Unary AAC decoding	bIsUnaryAACs are enabled for all components	oi_kp#4-unaryaac	oi_kp#4-unaryaac
oi_kp#4-7	the flag, indicating whether the number of key values is more than 2	bIsMoreTwoKVs is enabled	oi_kp#4-bIsMoreTwoKVs	oi_kp#4-bIsMoreTwoKVs

8.5.2.3.2.4 Test Bitstreams for path preserving

♦ **Test Bitstreams pathpreserving#1**

Purpose: Exercise the path preserving functionality of MPEG-4 Orientation Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
oi_pp#1-1	key quantization bit size decoding	nKeyQBit is set to 10	oi_pp#1-nKQBit	oi_pp#1-nKQBit
oi_pp#1-2	linear key decoding	bIsLinearSubRegion is enabled	oi_pp#1-LinearKey	oi_pp#1-LinearKey
oi_pp#1-3	linear key decoding	bIsLinearSubRegion is disabled	oi_pp#1-NoLinearKey	oi_pp#1-NoLinearKey

oi_pp#1-4	key range decoding	bRangeFlag is enabled	oi_pp#1-OverKeyRange	oi_pp#1-OverKeyRange
oi_pp#1-5	key range decoding	bRangeFlag is disabled	oi_pp#1-KeyRange	oi_pp#1-KeyRange
oi_pp#1-6	n-th order DPCM	nDPCMOrder is 2nd order	oi_pp#1-2ndDPCM	oi_pp#1-2ndDPCM
oi_pp#1-7	shift operation	bShiftFlag is enabled	oi_pp#1-Shift	oi_pp#1-Shift
oi_pp#1-8	shift operation	bShiftFlag is disabled	oi_pp#1-NoShift	oi_pp#1-NoShift
oi_pp#1-9	if DND and folding operations are not selected	nDNDOrder is -1	oi_pp#1-MinusOneDNDOrder	oi_pp#1-MinusOneDNDOrder
oi_pp#1-10	folding operation	nDNDOrder is 0	oi_pp#1-ZeroDNDOrder	oi_pp#1-ZeroDNDOrder
oi_pp#1-11	DND operation after folding operation	nDNDOrder is 3	oi_pp#1-PositiveDNDOrder	oi_pp#1-PositiveDNDOrder
oi_pp#1-12	key Invert down operation	bKeyInvertDown Flag is enabled	oi_pp#1-KeyInvertDown	oi_pp#1-KeyInvertDown
oi_pp#1-13	SignedAAC decoding	bKeyInvertDown Flag is disabled when nDNDOrder is 1~7	oi_pp#1-NoKeyInvertDownAndSignedAAC	oi_pp#1-NoKeyInvertDownAndSignedAAC
oi_pp#1-14	UnsignedAAC decoding	bKeyInvertDown Flag is disabled when nDNDOrder is –1~0	oi_pp#1-NoKeyInvertDownAndUnsignedAAC	oi_pp#1-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams pathpreserving#2**

Purpose: Exercise the path preserving functionality of MPEG-4 Orientation Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
oi_pp#2-1	key value quantization bit size decoding	nKVQBit is set to 10	oi_pp#2-nKVQBit	oi_pp#2-nKVQBit
oi_pp#2-2	1st-order DPCM	nKVDPCMOrder is set to 0	oi_pp#2-1stDPCM	oi_pp#2-1stDPCM
oi_pp#2-3	2nd-order DPCM (with Circular DPCM)	nKVDPCMOrder is set to 1	oi_pp#2-2ndDPCM	oi_pp#2-2ndDPCM
oi_pp#2-4	key value flag, indicating whether all quantized values are same in the each component of key values	x_keyvalue_flag is enabled y_keyvalue_flag is disabled z_keyvalue_flag is enabled	oi_pp#2-allsame	oi_pp#2-allsame

oi_pp#2-5	Binary AAC decoding	bIsUnaryAACs are disabled for all components	oi_pp#2-binaryaac	oi_pp#2-binaryaac
oi_pp#2-6	Unary AAC decoding	bIsUnaryAACs are enabled for all components	oi_pp#2-unaryaac	oi_pp#2-unaryaac
oi_pp#2-7	the flag, indicating whether the number of key values is more than 2	bIsMoreTwoKVs is enabled	oi_pp#2-bIsMoreTwoKVs	oi_pp#2-bIsMoreTwoKVs

♦ **Test Bitstreams pathpreserving#3**

Purpose: Exercise the path preserving functionality of MPEG-4 Orientation Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
oi_pp#3-1	key quantization bit size decoding	nKeyQBit is set to 10	oi_pp#3-nKQBit	oi_pp#3-nKQBit
oi_pp#3-2	linear key decoding	bIsLinearSubRegion is enabled	oi_pp#3-LinearKey	oi_pp#3-LinearKey
oi_pp#3-3	linear key decoding	bIsLinearSubRegion is disabled	oi_pp#3-NoLinearKey	oi_pp#3-NoLinearKey
oi_pp#3-4	key range decoding	bRangeFlag is enabled	oi_pp#3-OverKeyRange	oi_pp#3-OverKeyRange
oi_pp#3-5	key range decoding	bRangeFlag is disabled	oi_pp#3-KeyRange	oi_pp#3-KeyRange
oi_pp#3-6	n-th order DPCM	nDPCMOrder is 2nd order	oi_pp#3-2ndDPCM	oi_pp#3-2ndDPCM
oi_pp#3-7	shift operation	bShiftFlag is enabled	oi_pp#3-Shift	oi_pp#3-Shift
oi_pp#3-8	shift operation	bShiftFlag is disabled	oi_pp#3-NoShift	oi_pp#3-NoShift
oi_pp#3-9	if DND and folding operations are not selected	nDNDOrder is -1	oi_pp#3-MinusOneDNDOrder	oi_pp#3-MinusOneDNDOrder
oi_pp#3-10	folding operation	nDNDOrder is 0	oi_pp#3-ZeroDNDOrder	oi_pp#3-ZeroDNDOrder
oi_pp#3-11	DND operation after folding operation	nDNDOrder is 3	oi_pp#3-PositiveDNDOrder	oi_pp#3-PositiveDNDOrder
oi_pp#3-12	key Invert down operation	bKeyInvertDownFlag is enabled	oi_pp#3-KeyInvertDown	oi_pp#3-KeyInvertDown
oi_pp#3-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	oi_pp#3-NoKeyInvertDownAndSignedAAC	oi_pp#3-NoKeyInvertDownAndSignedAAC
oi_pp#3-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	oi_pp#3-NoKeyInvertDownAndUnsignedAAC	oi_pp#3-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams pathpreserving#4**

Purpose: Exercise the path preserving functionality of MPEG-4 Orientation Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
oi_pp#4-1	key value quantization bit size decoding	nKVQBit is set to 10	oi_pp#4-nKVQBit	oi_pp#4-nKVQBit
oi_pp#4-2	1st-order DPCM	nKVDPCMOrder is set to 0	oi_pp#4-1stDPCM	oi_pp#4-1stDPCM
oi_pp#4-3	2nd-order DPCM (with Circular DPCM)	nKVDPCMOrder is set to 1	oi_pp#4-2ndDPCM	oi_pp#4-2ndDPCM
oi_pp#4-4	key value flag, indicating whether all quantized values are same in the each component of key values	x_keyvalue_flag is enabled y_keyvalue_flag is disabled z_keyvalue_flag is enabled	oi_pp#4-allsame	oi_pp#4-allsame
oi_pp#4-5	Binary AAC decoding	bIsUnaryAACs are disabled for all components	oi_pp#4-binaryaac	oi_pp#4-binaryaac
oi_pp#4-6	Unary AAC decoding	bIsUnaryAACs are enabled for all components	oi_pp#4-unaryaac	oi_pp#4-unaryaac
oi_pp#4-7	the flag, indicating whether the number of key values is more than 2	bIsMoreTwoKVs is enabled	oi_pp#4-bIsMoreTwoKVs	oi_pp#4-bIsMoreTwoKVs

8.5.2.3.2.5 Tolerance

The diagnosis is to check whether the field data (key, keyValue) of OrientationInterpolator node that is decoded from ".mp4" file correspond with the node included in the provided reference file.

8.5.2.3.3 Position Interpolator

8.5.2.3.3.1 Conformance Requirements

Since the functionalities are not directly supported by a profile@level combination, the conformance of the corresponding decoder shall be defined, such that the decoder conforms to the functionality.

A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-11:2004 and meets all the general requirements, defined in that document, which apply for the functionality considered, and if it can decode bitstreams with any options or parameters with values permitted for that functionality. In the case of using BIFS for scene representation, the decoding process that is specified in Clause 8 of ISO/IEC 14496-1:2004 shall also be implemented.

In the following subclauses the term 'reference decoder' means the technical report software verifier (ISO/IEC 14496-5:2001/Amd.7). The reference decoder is a decoder that implements precisely the decoding process as specified in ISO/IEC 14496-11:2004.

8.5.2.3.3.2 Measurement Procedure

To test the validity of the output of a position interpolator decoder which is not included in a profile@level combination, the conformance will be performed only for the functionality independent from the existing profiles. Every decoder shall be able to decode the bitstreams provided for key preserving and path-preserving functionalities as described in subclause 8.5.2.3.3.3 and 8.5.2.3.3.4

8.5.2.3.3.3 Test Bitstreams for key preserving

- **Test Bitstreams keypreserving#1**

Purpose: Exercise the key preserving functionality of MPEG-4 Position Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
pi_kp#1-1	key quantization bit size decoding	nKeyQBit is set to 10	pi_kp#1-nKQBit	pi_kp#1-nKQBit
pi_kp#1-2	linear key decoding	bIsLinearSubRegion is enabled	pi_kp#1-LinearKey	pi_kp#1-LinearKey
pi_kp#1-3	linear key decoding	bIsLinearSubRegion is disabled	pi_kp#1-NoLinearKey	pi_kp#1-NoLinearKey
pi_kp#1-4	key range decoding	bRangeFlag is enabled	pi_kp#1-OverKeyRange	pi_kp#1-OverKeyRange
pi_kp#1-5	key range decoding	bRangeFlag is disabled	pi_kp#1-KeyRange	pi_kp#1-KeyRange
pi_kp#1-6	n-th order DPCM	nDPCMOrder is 2nd order	pi_kp#1-2ndDPCM	pi_kp#1-2ndDPCM
pi_kp#1-7	shift operation	bShiftFlag is enabled	oi_kp#1-Shift	pi_kp#1-Shift
pi_kp#1-8	shift operation	bShiftFlag is disabled	pi_kp#1-NoShift	pi_kp#1-NoShift
pi_kp#1-9	if DND and folding operations are not selected	nDNDOrder is -1	pi_kp#1-MinusOneDNDOrder	pi_kp#1-MinusOneDNDOrder
pi_kp#1-10	folding operation	nDNDOrder is 0	pi_kp#1-ZeroDNDOrder	pi_kp#1-ZeroDNDOrder
pi_kp#1-11	DND operation after folding operation	nDNDOrder is 3	pi_kp#1-PositiveDNDOrder	pi_kp#1-PositiveDNDOrder
pi_kp#1-12	key Invert down operation	bKeyInvertDownFlag is enabled	pi_kp#1-KeyInvertDown	pi_kp#1-KeyInvertDown
pi_kp#1-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	pi_kp#1-NoKeyInvertDownAndSignedAAC	pi_kp#1-NoKeyInvertDownAndSignedAAC
pi_kp#1-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	pi_kp#1-NoKeyInvertDownAndUnsignedAAC	pi_kp#1-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams keypreserving#2**

Purpose: Exercise the key preserving functionality of MPEG-4 Position Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
pi_kp#2-1	key value quantization bit size decoding	nKVQBit is set to 10	pi_kp#2-nKVQBit	pi_kp#2-nKVQBit
pi_kp#2-2	n-th order DPCM	nKVDPCMOrders are set to 1 for all components	pi_kp#2-1stDPCM	pi_kp#2-1stDPCM
pi_kp#2-3	key value flag, indicating whether all quantized values are same in the each component of key values	x_keyvalue_flag is enabled, y_keyvalue_flag is enabled, z_keyvalue_flag is disabled	pi_kp#2-allsame	pi_kp#2-allsame
pi_kp#2-4	Binary AAC decoding	bIsUnaryAACs are disabled for all components	pi_kp#2-binaryaac	pi_kp#2-binaryaac
pi_kp#2-5	Unary AAC decoding	bIsUnaryAACs are enabled for all components	pi_kp#2-unaryaac	pi_kp#2-unaryaac
pi_kp#2-6	Intra value decoding	nStartIndex_X, nStartIndex_Y and nStartIndex_Z are set to 1	pi_kp#2-intracoding	pi_kp#2-intracoding
pi_kp#2-7	floating point number decoding, which consists of mantissa and exponent.	bUse32Float is enabled	pi_kp#2-FPNcoding	pi_kp#2-FPNcoding
pi_kp#2-8	the decoding of the digits of the minimum values of each component, when all these values have the different digit	bAllSameMantissa DigitFlag is disabled	pi_kp#2-allsamemantissadigit	pi_kp#2-allsamemantissadigit
pi_kp#2-9	the decoding of the digits of the minimum values of each component, when all these values have the same digit	bSameKVDigitFlag is disabled	pi_kp#2-samekvdigit	pi_kp#2-samekvdigit
pi_kp#2-10	the decoding of the digit of the maximum range among the ranges of each components	bMaxDigitFlag is disabled	pi_kp#2-maxdigit	pi_kp#2-maxdigit
pi_kp#2-11	the sign of exponent of all the minimum values and the maximum range	bAllSameExponent Sign is disabled	pi_kp#2-allsameexponentsign	pi_kp#2-allsameexponentsign
pi_kp#2-12	the decoding of all the minimum values and the maximum range	nDigit is set to 6	pi_kp#2-digit	pi_kp#2-digit

♦ **Test Bitstreams keypreserving#3**

Purpose: Exercise the key preserving functionality of MPEG-4 Position Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.txt - IM1 textual format)
pi_kp#3-1	key quantization bit size decoding	nKeyQBit is set to 10	pi_kp#3-nKQBit	pi_kp#3-nKQBit
pi_kp#3-2	linear key decoding	bIsLinearSubRegion is enabled	pi_kp#3-LinearKey	pi_kp#3-LinearKey
pi_kp#3-3	linear key decoding	bIsLinearSubRegion is disabled	pi_kp#3-NoLinearKey	pi_kp#3-NoLinearKey
pi_kp#3-4	key range decoding	bRangeFlag is enabled	pi_kp#3-OverKeyRange	pi_kp#3-OverKeyRange
pi_kp#3-5	key range decoding	bRangeFlag is disabled	pi_kp#3-KeyRange	pi_kp#3-KeyRange
pi_kp#3-6	n-th order DPCM	nDPCMOrder is 2nd order	pi_kp#3-2ndDPCM	pi_kp#3-2ndDPCM
pi_kp#3-7	shift operation	bShiftFlag is enabled	oi_kp#3-Shift	pi_kp#3-Shift
pi_kp#3-8	shift operation	bShiftFlag is disabled	pi_kp#3-NoShift	pi_kp#3-NoShift
pi_kp#3-9	if DND and folding operations are not selected	nDNDOrder is -1	pi_kp#3-MinusOneDNDOrder	pi_kp#3-MinusOneDNDOrder
pi_kp#3-10	folding operation	nDNDOrder is 0	pi_kp#3-ZeroDNDOrder	pi_kp#3-ZeroDNDOrder
pi_kp#3-11	DND operation after folding operation	nDNDOrder is 3	pi_kp#3-PositiveDNDOrder	pi_kp#3-PositiveDNDOrder
pi_kp#3-12	key Invert down operation	bKeyInvertDownFlag is enabled	pi_kp#3-KeyInvertDown	pi_kp#3-KeyInvertDown
pi_kp#3-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	pi_kp#3-NoKeyInvertDownAndSignedAAC	pi_kp#3-NoKeyInvertDownAndSignedAAC
pi_kp#3-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	pi_kp#3-NoKeyInvertDownAndUnsignedAAC	pi_kp#3-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams keypreserving#4**

Purpose: Exercise the key preserving functionality of MPEG-4 Position Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.txt - IM1 textual format)
pi_kp#4-1	key value quantization bit size decoding	nKVQBit is set to 10	pi_kp#4-nKVQBit	pi_kp#4-nKVQBit
pi_kp#4-2	n-th order DPCM	nKVDPCMOrders are set to 1 for all components	pi_kp#4-1stDPCM	pi_kp#4-1stDPCM
pi_kp#4-3	key value flag, indicating whether all quantized values are same in the each component of key values	x_keyvalue_flag is enabled, y_keyvalue_flag is enabled, z_keyvalue_flag is disabled	pi_kp#4-allsame	pi_kp#4-allsame
pi_kp#4-4	Binary AAC decoding	bIsUnaryAACs are disabled for all components	pi_kp#4-binaryaac	pi_kp#4-binaryaac
pi_kp#4-5	Unary AAC decoding	bIsUnaryAACs are enabled for all components	pi_kp#4-unaryaac	pi_kp#4-unaryaac
pi_kp#4-6	Intra value decoding	nStartIndex_X, nStartIndex_Y and nStartIndex_Z are set to 1	pi_kp#4-intracoding	pi_kp#4-intracoding
pi_kp#4-7	floating point number decoding, which consists of mantissa and exponent.	bUse32Float is enabled	pi_kp#4-FPNcoding	pi_kp#4-FPNcoding
pi_kp#4-8	the decoding of the digits of the minimum values of each component, when all these values have the different digit	bAllSameMantissaDigitFlag is disabled	pi_kp#4-allsamemantissadigit	pi_kp#4-allsamemantissadigit
pi_kp#4-9	the decoding of the digits of the minimum values of each component, when all these values have the same digit	bSameKVDigitFlag is disabled	pi_kp#4-samekvdigit	pi_kp#4-samekvdigit
pi_kp#4-10	the decoding of the digit of the maximum range among the ranges of each components	bMaxDigitFlag is disabled	pi_kp#4-maxdigit	pi_kp#4-maxdigit
pi_kp#4-11	the sign of exponent of all the minimum values and the maximum range	bAllSameExponentSign is disabled	pi_kp#4-allsameexponentsign	pi_kp#4-allsameexponentsign
pi_kp#4-12	the decoding of all the minimum values and the maximum range	nDigit is set to 6	pi_kp#4-digit	pi_kp#4-digit

8.5.2.3.3.4 Test Bitstreams for path preserving

♦ **Test Bitstreams pathpreserving#1**

Purpose: Exercise the path preserving functionality of MPEG-4 Position Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
pi_pp#1-1	key quantization bit size decoding	nKeyQBit is set to 10	pi_pp#1-nKQBit	pi_pp#1-nKQBit
pi_pp#1-2	linear key decoding	bIsLinearSubRegion is enabled	pi_pp#1-LinearKey	pi_pp#1-LinearKey
pi_pp#1-3	linear key decoding	bIsLinearSubRegion is disabled	pi_pp#1-NoLinearKey	pi_pp#1-NoLinearKey
pi_pp#1-4	key range decoding	bRangeFlag is enabled	pi_pp#1-OverKeyRange	pi_pp#1-OverKeyRange
pi_pp#1-5	key range decoding	bRangeFlag is disabled	pi_pp#1-KeyRange	pi_pp#1-KeyRange
pi_pp#1-6	n-th order DPCM	nDPCMOrder is 2nd order	pi_pp#1-2ndDPCM	pi_pp#1-2ndDPCM
pi_pp#1-7	shift operation	bShiftFlag is enabled	pi_pp#1-Shift	pi_pp#1-Shift
pi_pp#1-8	shift operation	bShiftFlag is disabled	pi_pp#1-NoShift	pi_pp#1-NoShift
pi_pp#1-9	if DND and folding operations are not selected	nDNDOrder is -1	pi_pp#1-MinusOneDNDOrder	pi_pp#1-MinusOneDNDOrder
pi_pp#1-10	folding operation	nDNDOrder is 0	pi_pp#1-ZeroDNDOrder	pi_pp#1-ZeroDNDOrder
pi_pp#1-11	DND operation after folding operation	nDNDOrder is 3	pi_pp#1-PositiveDNDOrder	pi_pp#1-PositiveDNDOrder
pi_pp#1-12	key Invert down operation	bKeyInvertDownFlag is enabled	pi_pp#1-KeyInvertDown	pi_pp#1-KeyInvertDown
pi_pp#1-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	pi_pp#1-NoKeyInvertDownAndSignedAAC	pi_pp#1-NoKeyInvertDownAndSignedAAC
pi_pp#1-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	pi_pp#1-NoKeyInvertDownAndUnsignedAAC	pi_pp#1-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams pathpreserving#2**

Purpose: Exercise the path preserving functionality of MPEG-4 Position Interpolator Compression carried in a separate stream from the scene stream (BIFS stream). The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
pi_pp#2-1	key value quantization bit size decoding	nKVQBit is set to 10	pi_pp#2-nKVQBit	pi_pp#2-nKVQBit
pi_pp#2-2	n-th order DPCM	nKVDPCMOrders are set to 1 for all components	pi_pp#2-1stDPCM	pi_pp#2-1stDPCM
pi_pp#2-3	key value flag, indicating whether all quantized values are same in the each component of key values	x_keyvalue_flag is enabled, y_keyvalue_flag is enabled, z_keyvalue_flag is disabled	pi_pp#2-allsame	pi_pp#2-allsame
pi_pp#2-4	Binary AAC decoding	bIsUnaryAACs are disabled for all components	pi_pp#2-binaryaac	pi_pp#2-binaryaac
pi_pp#2-5	Unary AAC decoding	bIsUnaryAACs are enabled for all components	pi_pp#2-unaryaac	pi_pp#2-unaryaac
pi_pp#2-6	Intra value decoding	nStartIndex_X, nStartIndex_Y and nStartIndex_Z are set to 1	pi_pp#2-intracoding	pi_pp#2-intracoding
pi_pp#2-7	floating point number decoding, which consists of mantissa and exponent.	bUse32Float is enabled	pi_pp#2-FPNcoding	pi_pp#2-FPNcoding
pi_pp#2-8	the decoding of the digits of the minimum values of each component, when all these values have the different digit	bAllSameMantissaDigitFlag is disabled	pi_pp#2-allsamemantissadigit	pi_pp#2-allsamemantissadigit
pi_pp#2-9	the decoding of the digits of the minimum values of each component, when all these values have the same digit	bSameKVDigitFlag is disabled	pi_pp#2-samekvdigit	pi_pp#2-samekvdigit
pi_pp#2-10	the decoding of the digit of the maximum range among the ranges of each components	bMaxDigitFlag is disabled	pi_pp#2-maxdigit	pi_pp#2-maxdigit
pi_pp#2-11	the sign of exponent of all the minimum values and the maximum range	bAllSameExponentSign is disabled	pi_pp#2-allsameexponentsign	pi_pp#2-allsameexponentsign
pi_pp#2-12	the decoding of all the minimum values and the maximum range	nDigit is set to 6	pi_pp#2-digit	pi_pp#2-digit

♦ **Test Bitstreams pathpreserving#3**

Purpose: Exercise the path preserving functionality of MPEG-4 Position Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key data.

File :

Test Name	Attribute	Bitstream info. For key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
pi_pp#3-1	key quantization bit size decoding	nKeyQBit is set to 10	pi_pp#3-nKQBit	pi_pp#3-nKQBit
pi_pp#3-2	linear key decoding	bIsLinearSubRegion is enabled	pi_pp#3-LinearKey	pi_pp#3-LinearKey
pi_pp#3-3	linear key decoding	bIsLinearSubRegion is disabled	pi_pp#3-NoLinearKey	pi_pp#3-NoLinearKey
pi_pp#3-4	key range decoding	bRangeFlag is enabled	pi_pp#3-OverKeyRange	pi_pp#3-OverKeyRange
pi_pp#3-5	key range decoding	bRangeFlag is disabled	pi_pp#3-KeyRange	pi_pp#3-KeyRange
pi_pp#3-6	n-th order DPCM	nDPCMOrder is 2nd order	pi_pp#3-2ndDPCM	pi_pp#3-2ndDPCM
pi_pp#3-7	shift operation	bShiftFlag is enabled	pi_pp#3-Shift	pi_pp#3-Shift
pi_pp#3-8	shift operation	bShiftFlag is disabled	pi_pp#3-NoShift	pi_pp#3-NoShift
pi_pp#3-9	if DND and folding operations are not selected	nDNDOrder is -1	pi_pp#3-MinusOneDNDOrder	pi_pp#3-MinusOneDNDOrder
pi_pp#3-10	folding operation	nDNDOrder is 0	pi_pp#3-ZeroDNDOrder	pi_pp#3-ZeroDNDOrder
pi_pp#3-11	DND operation after folding operation	nDNDOrder is 3	pi_pp#3-PositiveDNDOrder	pi_pp#3-PositiveDNDOrder
pi_pp#3-12	key Invert down operation	bKeyInvertDownFlag is enabled	pi_pp#3-KeyInvertDown	pi_pp#3-KeyInvertDown
pi_pp#3-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	pi_pp#3-NoKeyInvertDownAndSignedAAC	pi_pp#3-NoKeyInvertDownAndSignedAAC
pi_pp#3-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	pi_pp#3-NoKeyInvertDownAndUnsignedAAC	pi_pp#3-NoKeyInvertDownAndUnsignedAAC

♦ **Test Bitstreams pathpreserving#4**

Purpose: Exercise the path preserving functionality of MPEG-4 Position Interpolator Compression carried in a BIFS scene stream. The test is performed for each type of attribute related to the key value data.

File :

Test Name	Attribute	Bitstream info. For key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
pi_pp#4-1	key value quantization bit size decoding	*nKVQBit* is set to 10	pi_pp#4-nKVQBit	pi_pp#4-nKVQBit
pi_pp#4-2	n-th order DPCM	*nKVDPCMOrder*s are set to 1 for all components	pi_pp#4-1stDPCM	pi_pp#4-1stDPCM
pi_pp#4-3	key value flag, indicating whether all quantized values are same in the each component of key values	*x_keyvalue_flag* is enabled, *y_keyvalue_flag* is enabled, *z_keyvalue_flag* is disabled	pi_pp#4-allsame	pi_pp#4-allsame
pi_pp#4-4	Binary AAC decoding	*bIsUnaryAAC*s are disabled for all components	pi_pp#4-binaryaac	pi_pp#4-binaryaac
pi_pp#4-5	Unary AAC decoding	*bIsUnaryAAC*s are enabled for all components	pi_pp#4-unaryaac	pi_pp#4-unaryaac
pi_pp#4-6	Intra value decoding	*nStartIndex_X*, *nStartIndex_Y* and *nStartIndex_Z* are set to 1	pi_pp#4-intracoding	pi_pp#4-intracoding
pi_pp#4-7	floating point number decoding, which consists of mantissa and exponent.	*bUse32Float* is enabled	pi_pp#4-FPNcoding	pi_pp#4-FPNcoding
pi_pp#4-8	the decoding of the digits of the minimum values of each component, when all these values have the different digit	*bAllSameMantissaDigitFlag* is disabled	pi_pp#4-allsamemantissadigit	pi_pp#4-allsamemantissadigit
pi_pp#4-9	the decoding of the digits of the minimum values of each component, when all these values have the same digit	*bSameKVDigitFlag* is disabled	pi_pp#4-samekvdigit	pi_pp#4-samekvdigit
pi_pp#4-10	the decoding of the digit of the maximum range among the ranges of each components	*bMaxDigitFlag* is disabled	pi_pp#4-maxdigit	pi_pp#4-maxdigit
pi_pp#4-11	the sign of exponent of all the minimum values and the maximum range	*bAllSameExponentSign* is disabled	pi_pp#4-allsameexponentsign	pi_pp#4-allsameexponentsign
pi_pp#4-12	the decoding of all the minimum values and the maximum range	*nDigit* is set to 6	pi_pp#4-digit	pi_pp#4-digit

8.5.2.3.3.5 Tolerance

The diagnosis is to check whether the field data (key, keyValue) of PositionInterpolator node that is decoded from ".mp4" file correspond with the node included in the provided reference file.

8.5.3 MeshGrid

MESHGRID surface representation describes the model both as a mesh – the connectivity-wireframe, and as a volume surrounding the mesh – the reference-grid, as described in subclause 4.3.4 of ISO/IEC 14496-16:2004. The fields of the MeshGrid node are encoded either by BIFS, or in a compressed representation as a binary stream as explained in subclause 5.2 of ISO/IEC 14496-16:2004. In both cases the information needs to be decoded to obtain the model as given in subclause 5.2.3 of ISO/IEC 14496-16:2004. The animation of the mesh is done either by changing the coordinates of the vertices, or indirectly by changing the coordinates of the reference-grid points or the vertices offsets as described in subclause 4.3.4.3 of ISO/IEC 14496-16:2004.

8.5.3.1 Conformance Points

8.5.3.1.1 Covered functionalities

The conformance points for MeshGrid covers the decoding of the BIFS encoded fields, the decoding of the compressed representation, the animation of the reference-grid points, and the animation of the vertices offsets to achieve the movement of the vertices. The compressed representation is carried by BitWrapper node as described in ISO/IEC 14496-11:2004. The animation functionality is independent of the encoding method of the fields, whether they are encoded by BIFS or by the compressed representation.

As for carriage of compressed representation of MeshGrid node using BitWrapper node, it can be carried either in a separate stream (outband scenario) or within the scene stream, the BIFS stream (inband scenario). Therefore, MeshGrid also shall be tested together with this node as described in ISO/IEC 14496-11:2004.

The following subclauses specify the normative tests for verifying conformance of MeshGrid bitstream and MeshGrid decoder. Those normative tests make use of test data (bitstream test suites) provided as an electronic annex to this document, and of a software verifier specified in ISO/IEC 14496-5:2001/Amd.7 with source code available in electronic format.

8.5.3.2 Bitstream Conformance

8.5.3.2.1 Conformance Requirements

BIFS streams shall comply with the specifications in subclause 5.2 of ISO/IEC 14496-16:2004.

8.5.3.2.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of subclause 5.2 of ISO/IEC 14496-16:2004.

8.5.3.2.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

8.5.3.3 Terminal conformance

8.5.3.3.1 Conformance Requirements

The terminal shall decode the BIFS stream and initialize the fields of the object of subclause 4.3.4 of ISO/IEC 14496-16:2004.

8.5.3.3.2 Test Bitstreams

8.5.3.3.2.1 Test Bitstreams for encoding with BIFS

In the case of using BIFS for scene representation for encoding the fields of the MeshGrid node, a decoding is required, and the appropriate fields of the MeshGrid node are initialized, as specified in subclause 4.3.4.2 of ISO/IEC 14496-16:2004.

Test Name	Provider	Description	File (*.mp4)	Reference File (*.xml)
Torus_C_LA_ BIFS	V.U.B.	Multi-resolution non-homogeneous mesh with uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field	Torus_C_LA_BIF S	Torus_BIFS
Torus_C_OA_ BIFS	V.U.B.	Multi-resolution non-homogeneous mesh with uniform distributed reference grid. Animation of the vertexOffset field.	Torus_C_OA_BI FS	Torus_BIFS
Humanoid_LA_ BIFS	V.U.B.	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field	Humanoid_LA_BI FS	Humanoid_BIFS
Humanoid_OA_ BIFS	V.U.B.	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid. Animation of the vertexOffset field.	Humanoid_OA_B IFS	Humanoid_BIFS
Sphere_GA_ BIFS	V.U.B.	Multi-resolution non-homogeneous mesh. Animation of the gridCoord field.	Sphere_GA_BIF S	Sphere_BIFS
Quad_LA_BIFS	V.U.B.	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field	Quad_LA_BIFS	Quad_BIFS
Quad_OA_BIFS	V.U.B.	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Animation of the vertexOffset field.	Quad_OA_BIFS	Quad_BIFS
Quad_GA_BIFS	V.U.B.	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Animation of the gridCoord field.	Quad_GA_BIFS	Quad_BIFS
Cyclic_LA_BIFS	V.U.B.	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field	Cyclic_LA_BIFS	Cyclic_BIFS
Cyclic_OA_BIFS	V.U.B.	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Animation of the vertexOffset field.	Cyclic_OA_BIFS	Cyclic_BIFS
Cyclic_GA_BIFS	V.U.B.	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Animation of the gridCoord field.	Cyclic_GA_BIFS	Cyclic_BIFS

8.5.3.3.2.2 Test Bitstreams for outband encoding

The outband scenario shall be tested with the BitWrapper node. The bitstream is specified in the *url* field of the BitWrapper node. The animation functionality is the same as in the case of using BIFS for scene representation for encoding the fields as specified in subclause 8.5.3.3.2.1.

Test Name	Provider	Description	File (*.mp4)	Reference File (*.xml)
Torus_C_URL	V.U.B.	Multi-resolution non-homogeneous mesh with uniform distributed reference grid.	Torus_C_URL	Torus_MG.
Humanoid_URL	V.U.B.	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid.	Humanoid_URL	Humanoid_MG
Quad_URL	V.U.B.	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1).	Quad_URL	Quad_MG
Cyclic_URL	V.U.B.	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1).	Cyclic_URL	Cyclic_MG

8.5.3.3.2.3 Test Bitstreams for inband encoding

The inband scenario shall be tested with the BitWrapper node. The bitstream is specified in the *buffer* field of the BitWrapper node. The animation functionality is the same as in the case of using BIFS for scene representation for encoding the fields as specified in subclause 8.5.3.3.2.1.

Test Name	Provider	Description	File (*.mp4)	Reference File (*.xml)
Torus_C_LA_BUFFER	V.U.B.	Multi-resolution non-homogeneous mesh with uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field	Torus_C_LA_BUFFER	Torus_MG.
Humanoid_LA_BUFFER	V.U.B.	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field	Humanoid_LA_BUFFER	Humanoid_MG
Quad_LA_BUFFER	V.U.B.	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field	Quad_LA_BUFFER	Quad_MG
Cyclic_LA_BUFFER	V.U.B.	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field	Cyclic_LA_BUFFER	Cyclic_MG

8.5.3.3.3 Measurement Procedure

The terminal shall produce the rendered image and a formatted output giving the reconstructed fields of the MeshGrid object from the bitstream as described in subclause 8.5.3.3.2. The decoder shall be able to decode the bitstreams provided as described in subclause 8.5.3.3.2. The output should include the description of the following fields of the node description defined in subclause 4.3.4 of ISO/IEC 14496-16:2004:

- The fields from the DecoderSpecificInfo of MeshGrid in the *DecoderInfo* section.
- The *gridCoord* field of MeshGrid as a formatted output in the *ReferenceGrid* section.
- The *coord*, *vertexLink* and *vertexOffset* fields as a formatted output, in the *ConnectivityWireframe* section.
- The *coordIndex* field as a formatted output in the *Triangles* section.

8.5.3.3.4 Output Format

The output format is in XML. An example is the following:

```
<?xml version="1.0"?>

<MeshGrid>
     <DecoderInfo
         levelsU="5" levelsV="5" levelsW="5"
         slicesU="49" slicesV="49" slicesW="33"
         HasConnectivityInfo="true" HasRefinementInfo="true"
         HasRepositionInfo="true" HasGridInfo="false"
         MeshType="Generic" SameBorderDirection="false" UniformSplit="false"
         cyclicU="false" cyclicV="false" cyclicW="false"
         refineBits="1" fullRefine="false"
         gridScaleX="-1" gridScaleY="-1" gridScaleZ="-1">
     </DecoderInfo>

     <ReferenceGrid>
         <depth w="0">
            <width v="0">
               <height u="0" x="-2.000" y="-2.000" z="-1.000"></height>
               <height u="1" x="-1.083" y="-2.000" z="-1.000"></height>
               <height u="2" x="-0.167" y="-2.000" z="-1.000"></height>            ...
            </width>
            ...
            <width v="15">
               <height u="0" x="-2.000" y="11.750" z="-1.000"></height>
               <height u="1" x="-1.083" y="11.750" z="-1.000"></height>
               <height u="2" x="-0.167" y="11.750" z="-1.000"></height>            ...
            </width>
         </depth>
         ...
         <depth w="21">
            <width v="0">
               <height u="0" x="-2.000" y="-2.000" z="13.438"></height>
               <height u="1" x="-1.083" y="-2.000" z="13.438"></height>
               <height u="2" x="-0.167" y="-2.000" z="13.438"></height>            ...
            </width>
            ...
            <width v="15">
               <height u="0" x="-2.000" y="11.750" z="13.438"></height>
               <height u="1" x="-1.083" y="11.750" z="13.438"></height>
               <height u="2" x="-0.167" y="11.750" z="13.438"></height>            ...
            </width>
         </depth>
     </ReferenceGrid>
```

```
      <ConnectivityWireframe>
          <resolution level="0">
             <vertex index="0"
                u="5" v="16" w="16"
                offset="0.865" borderDir="BACK"
                x="1.790" y="12.667" z="10.000"
                linkNext1="1" linkNext2="2" linkPrev1="6" linkPrev2="11">
             </vertex>
             ...
             <vertex index="12"
                u="42" v="32" w="16"
                offset="0.317" borderDir="FRONT"
                x="36.790" y="27.333" z="10.000"
                linkNext1="15" linkNext2="9" linkPrev1="8" linkPrev2="7">
             </vertex>
          </resolution>
          ...
          <resolution level="4">
             <vertex index="1725"
                u="3" v="21" w="17"
                offset="0.125" borderDir="LEFT"
                x="0.750" y="17.135" z="10.688"
                linkNext1="1674" linkNext2="1676" linkPrev1="1716" linkPrev2="2073">
             </vertex>
             ...
             <vertex index="375"
                u="44" v="26" w="16"
                offset="0.042" borderDir="RIGHT"
                x="38.333" y="21.872" z="10.000"
                linkNext1="1597" linkNext2="6377" linkPrev1="6376" linkPrev2="1619">
             </vertex>
          </resolution>
      </ConnectivityWireframe>

      <Triangles>
          <resolution level="0">
             <triangle index1="0" index2="2" index3="6"></triangle>
             <triangle index1="0" index2="1" index3="4"></triangle>
             <triangle index1="0" index2="4" index3="2"></triangle>
             ...
          </resolution>
          ...
          <resolution level="4">
             <triangle index1="1725" index2="1674" index3="1676"></triangle>
             <triangle index1="1716" index2="1725" index3="1676"></triangle>
             <triangle index1="1716" index2="1676" index3="1679"></triangle>
             ...
          </resolution>
      </Triangles>
  </MeshGrid>
```

The *ReferenceGrid* section lists the grid points, and the values for (u, v, w) represent the position of a grid point in the grid array, while the values for (x, y, z) represent the coordinates of the grid point, as defined in subclause 4.3.4.1.1 of ISO/IEC 14496-16:2004.

The *ConnectivityWireframe* section lists, for each resolution level, the vertices in the order they are decoded and their attributes as defined in the subclauses from below of ISO/IEC 14496-16:2004: (1) the order in the list (*index*), (2) the position (u, v, w) of a grid point to which the vertex is attached (subclause 4.3.4.3.2), (3) the offset (*offset*) (subclause 4.3.4.3.2 and 5.2.3.3), (4) the border direction (*borderDir*) (subclause 4.3.4.3.2 and 5.2.3.1), (5) the coordinates (*x*, *y*, *z*) (subclause 4.3.4.3.2 and 5.2.3.1), and (6) the two connectivity vectors (subclause 5.2.3.1).

The *Triangles* section gives, for each resolution level, the list of triangles where *index1*, *index2* and *index3* are the indices of the vertices from the *ConnectivityWireframe* section. The order of the triangles in the list is not imposed.

8.5.3.3.5 Tolerance

The rendering functionality must be observed visually. The conformance is passed when the output of (1) the object has the same values as the sample, (2) the coordinates (*x*, *y*, *z*) of the grid points from the *gridDescription* and of the vertices from the *vertexDescription* have a maximum error of $1/\text{scale}_{(x,y,z)}$ compared to the sample values, where $\text{scale}_{(x,y,z)}$ are the scaling factors (*gridScale*) defined for each of the x, y, z grid coeficients as specified in subclause 5.2.2.2 of ISO/IEC 14496-16:2004, (3) the remaining fields from the *vertexDescription* should have the same values as the sample, and (4) the *triangleDescription* contains all the entries in any order.

8.5.4 Depth Image-based representation

Depth image-based representations (DIBR) describe a new family of 3D representations for computer graphics and animation. Idea of the approach is to build a compact and photo-realistic representation of a 3D object or scene without using polygonal mesh. Instead, images accompanied by depth values for each pixel are used. This type of representation allows us to build and render novel views of photo-realistic objects and scene without using tremendous number of polygons. The main formats of the DIBR family are SimpleTexture (an image together with depth array), PointTexture (an image with multiple pixels along each line of sight), and OctreeImage (octree-like data structure together with a set of images containing viewport parameters).

8.5.4.1 Conformance Points

8.5.4.1.1 Covered functionalities

The conformance points cover the decoding of the BIFS encoded fields and the compressed representation of OctreeImage node in DIBR formats. The compressed representation is carried by BitWrapper node as described in ISO/IEC 14496-11:2004.

As for carriage of compressed representation of OctreeImage node using BitWrapper node, it can be carried either in a separate stream or within the scene stream (BIFS stream). Therefore, OctreeImage shall be tested together with this node as described in ISO/IEC 14496-11:2004.

The following subclauses specify the normative tests for verifying conformance of OctreeImage bitstream and decoder. Those normative tests make use of test data (bitstream test suites) provided as an electronic annex to this document, and of a software verifier specified in ISO/IEC 14496-5:2001/Amd.7 with source code available in electronic format.

8.5.4.2 Bitstream conformance

8.5.4.2.1 Conformance Requirements

BIFS streams shall comply with the specifications in subclause 5.3 of ISO/IEC 14496-16:2004 and subclause 7.2.2.20 of ISO/IEC 14496-11:2004.

8.5.4.2.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of subclause 5.3 of ISO/IEC 14496-16:2004 and subclause 7.2.2.20 of ISO/IEC 14496-11:2004.

8.5.4.2.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

8.5.4.3 Terminal conformance

8.5.4.3.1 Conformance Requirements

A decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-16:2004 and meets all the general requirements, defined in the document, which apply for the functionalities considered. The decoder shall decode bitstreams with any options or parameters with values permitted for the functionalities. In the case of using BIFS for scene representation, the decoding process that is specified in Clause 8 of ISO/IEC 14496-11:2004 shall also be implemented.

8.5.4.3.2 Measurement Procedure

The terminal should produce a formatted output giving the reconstructed fields of OctreeImage. The decoder shall be able to decode the bitstreams provided as described in subclause 8.5.4.3.4.

8.5.4.3.3 Tolerance

The diagnosis is to check whether the field data (octree and voxelImageIndex) of OctreeImage node decoded from ".mp4" files correspond with the node included in the provided reference (".ref") files.

8.5.4.3.4 Test Bitstreams

Test Name	Attributes	Bitstream (.mp4)	Reference file (.ref - DIBR only)
DIBR-OI-#1	Use of BitWrapper node. Use of URL field in the Bitwrapper node. Non-use of voxelImageIndex. Still version of OctreeImage node.	OI_Bitwrapper-URL__BVO_Still	OI_Bitwrapper-URL__BVO_Still
DIBR-OI-#2	Use of BitWrapper node. Use of URL field in the Bitwrapper node. Non-use of voxelImageIndex. Animated version of OctreeImage node.	OI_Bitwrapper-URL_BVO_Anim	OI_Bitwrapper-URL_BVO_Anim
DIBR-OI-#3	Use of BitWrapper node. Use of buffer field in the Bitwrapper node. Non-use of voxelImageIndex. Still version of OctreeImage node.	OI_Bitwrapper-Buffer_BVO_Still	OI_Bitwrapper-Buffer_BVO_Still
DIBR-OI-#4	Use of BitWrapper node. Use of buffer field in the Bitwrapper node. Non-use of voxelImageIndex. Animated version of OctreeImage node.	OI_ Bitwrapper - Buffer_BVO_Anim	OI_ Bitwrapper - Buffer_BVO_Anim
DIBR-OI-#5	Use of BitWrapper node. Use of URL field in the BitWrapper node. Use of voxelImageIndex. Still version of OctreeImage node.	OI_Bitwrapper-URL_TBVO_Still	OI_Bitwrapper-URL_TBVO_Still
DIBR-OI-#6	Use of BitWrapper node. Use of URL field in the BitWrapper node. Use of voxelImageIndex. Animated version of OctreeImage node.	OI_Bitwrapper-URL_TBVO_Anim	OI_Bitwrapper-URL_TBVO_Anim
DIBR-OI-#7	Use of BitWrapper node. Use of buffer field in the Bitwrapper node. Use of voxelImageIndex. Still version of OctreeImage node.	OI_Bitwrapper-Buffer_TBVO_Still	OI_Bitwrapper-Buffer_TBVO_Still
DIBR-OI-#8	Use of BitWrapper node. Use of buffer field in the Bitwrapper node. Use of voxelImageIndex. Animated version of OctreeImage node.	OI_Bitwrapper-Buffer_TBVO_Anim	OI_Bitwrapper-Buffer_TBVO_Anim

8.5.5 Wavelet Subdivision Surfaces

8.5.5.1 Bitstream conformance

8.5.5.1.1 Conformance Requirements

BIFS streams shall comply with the specifications in subclause 5.1 of ISO/IEC 14496-16:2004.

8.5.5.1.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of subclause 5.1 of ISO/IEC 14496-16:2004.

8.5.5.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

8.5.5.2 Terminal conformance

8.5.5.2.1 Conformance Requirements

WaveletSubdivisionSurface conformance points are the following:

- The ability to specify the number of components used in the reconstruction process;
- The ability to specify whether the wavelet coefficients are in local or global coordinates;
- The ability to specify the number of bitplanes used to decode each coordinate;
- The ability to specify which part of the mesh is to be reconstructed with a given coefficients packet;

8.5.5.2.2 Test Bitstreams

Name	Provider	Content	Original wrl file
WS_CHG	France Telecom R & D	WaveletSubdivisionSurface with complete spatial encoding, same quantization for each component and global coordinates for wavelet coefficients.	WS_CHG.wrl
WS_PNHL	France Telecom R & D	WaveletSubdivisionSurface with partial spatial encoding, different quantization for each component and local coordinates for wavelet coefficients.	WS_PNHL.wrl

8.5.5.2.3 Measurement Procedure

The terminal should produce a formatted output giving the reconstructed mesh with the available decoded wavelet coefficients. The output is to include

- The WaveletSubdivisionSurface object
- The **Basemesh** field.

8.5.5.2.4 Tolerance

The conformance is passed when the reconstructed mesh reflects the correct mesh, as defined in ISO/IEC 14496-16:2004. The rendering functionality must be observed visually.

8.5.6 Compressed Bone-based animation

In order to animate a skinned model by using the bone-based approach, a dedicated animation stream has been standardized. This information consists mainly in geometric transformation of the bones and curve control curve parameters. There are two ways to specify this data: textual and binary. While the textual format is informative and used for authoring purposes, the binary format is normative. Below there are conformance points concerning the compressed BBA elementary stream.

BBA compressed animation can be carried either in a separate stream or within the scene stream (BIFS stream). The following subclauses specify the normative tests for verifying conformance of BBA elementary bitstream and BBA decoder. Those normative tests make use of test data (bitstream test suites) provided as an electronic annex to this document, and of a software verifier specified in ISO/IEC 14496-5:2001/Amd.7 with source code available in electronic format.

8.5.6.1 Bitstream conformance

8.5.6.1.1 Conformance Requirements

BIFS streams shall comply with the specifications in subclause 5.4 of ISO/IEC 14496-16:2004.

8.5.6.1.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of subclause 5.4 of ISO/IEC 14496-16:2004.

8.5.6.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

8.5.6.2 Terminal conformance

There are two compression techniques that can be used in a BBA stream: frame-based and DCT-based. The following table presents the bitstreams for different use cases.

Test Name	Attributes	Elementary bitstream (.bba)	Bitstream (.mp4)
BBA_Bones_DCT01	Contains animation only for the bones. Compressed by DCT with quantization step 1.	BBA_Bones_DCT 01.bba	BBA_Bones_DCT01. mp4
BBA_Bones_DCT12	Contains animation only for the bones. Compressed by DCT with quantization step 12.	BBA_Bones_DCT 12.bba	BBA_Bones_DCT12. mp4
BBA_Bones_DCT31	Contains animation only for the bones. Compressed by DCT with quantization step 31.	BBA_Bones_DCT 31.bba	BBA_Bones_DCT31. mp4
BBA_Bones_PRE01	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 1.	BBA_Bones_PRE 01.bba	BBA_Bones_PRE01. mp4
BBA_Bones_PRE12	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 12.	BBA_Bones_PRE 12.bba	BBA_Bones_PRE12. mp4

BBA_Bones_PRE31	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 31.	BBA_Bones_PRE 31.bba	BBA_Bones_PRE31. mp4
BBA_BoneAnd Muscle_DCT01	Contains animation for bones and muscles. Compressed by DCT with quantization step 1.	BBA_BM_DCT01. bba	BBA_BM_DCT01. mp4
BBA_BoneAnd Muscle _DCT12	Contains animation for bones and muscles. Compressed by DCT with quantization step 12.	BBA_BM_DCT12. bba	BBA_BM_DCT12. mp4
BBA_BoneAnd Muscle _DCT31	Contains animation for bones and muscles. Compressed by DCT with quantization step 31.	BBA_BM_DCT31. bba	BBA_BM_DCT31. mp4
BBA_BoneAnd Muscle_PRE01	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 1.	BBA_BM_PRE01. bba	BBA_BM_PRE01. mp4
BBA_BoneAnd Muscle_PRE12	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 12.	BBA_BM_PRE12. bba	BBA_BM_PRE12. mp4
BBA_BoneAnd Muscle_PRE31	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 31.	BBA_BM_PRE31. bba	BBA_BM_PRE31. mp4

8.5.7 AFX Generic Backchannel

8.5.7.1 Bitstream conformance

8.5.7.1.1 Conformance Requirements

BIFS streams shall comply with the specifications in subclause 5.5 of ISO/IEC 14496-16:2004.

8.5.7.1.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of subclause 5.5 of ISO/IEC 14496-16:2004.

8.5.7.1.3 Tolerance

There is no tolerance for bitstream syntax checking. The diagnosis is pass or fail.

8.5.7.2 Terminal conformance

8.5.7.2.1 Conformance Requirements

BIFS streams shall comply with the specifications in subclause 5.5 of ISO/IEC 14496-16:2004.

8.5.7.2.2 Measurement Procedure

The terminal shall produce the rendered image and a formatted output of the object from the decoded stream. Because the backchannel supports view-dependent decoding, the correct rendered image and formatted output of the object should be produced from a partially decoded stream, taking only those stream packets that actively contribute to the visible object portions.

8.5.7.2.3 Output Format

The output format is in XML. An example is the following:

```
<BackChannelCommand tag="0x01">
     <fieldOfView x="", y="">
     <clipPlane near="", far="">
     <visibility limit="">
</BackChannelCommand>

<BackChannelCommand tag="0x02">
     <position x="", y="", z="">
     <direction x="", y="", z="", axisAngle="">
</BackChannelCommand>
```

8.5.7.2.4 Tolerance

The same rendered image can be produced in view-dependent mode by decoding a number of bitstream packets, ranging from only those that actively contribute to the visible object portions to all packets of the bitstream. This gives some freedom in the selection procedure of the packets. However, the rendered image should always be the same as if the object were rendered using all bitstream packets.

After decoding the packets corresponding to a series of viewpoints chosen in such a way that the entire object is covered, the output result should be the same as the one obtained after decoding the entire bitstream in one step.

ICS 35.040

Price based on 47 pages