INTERNATIONAL STANDARD

ISO/IEC 14496-27

First edition
2009-12-15

Information technology — Coding of audio-visual objects —

Part 27: 3D Graphics conformance

Technologies de l'information — Codage des objets audiovisuels —

Partie 27: Conformité aux graphiques 3D

Reference number
ISO/IEC 14496-27:2009(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

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

Contents

Page

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.

ISO/IEC 14496-27 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 14496 consists of the following parts, under the general title *Information technology — Coding of audio-visual objects*:

- *Part 1: Systems*
- *Part 2: Visual*
- *Part 3: Audio*
- *Part 4: Conformance testing*
- *Part 5: Reference software*
- *Part 6: Delivery Multimedia Integration Framework (DMIF)*
- *Part 7: Optimized reference software for coding of audio-visual objects*
- *Part 8: Carriage of ISO/IEC 14496 contents over IP networks*
- *Part 9: Reference hardware description*
- *Part 10: Advanced Video Coding*
- *Part 11: Scene description and application engine*
- *Part 12: ISO base media file format*
- *Part 13: Intellectual Property Management and Protection (IPMP) extensions*
- *Part 14: MP4 file format*

- *Part 15: Advanced Video Coding (AVC) file format*
- *Part 16: Animation Framework eXtension (AFX)*
- *Part 17: Streaming text format*
- *Part 18: Font compression and streaming*
- *Part 19: Synthesized texture stream*
- *Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)*
- *Part 21: MPEG-J Graphics Framework eXtensions (GFX)*
- *Part 22: Open Font Format*
- *Part 23: Symbolic Music Representation*
- *Part 24: Audio and systems interaction*
- *Part 25: 3D Graphics Compression Model*
- *Part 27: 3D Graphics conformance*

The following part is under preparation:

- Part 26: Audio conformance

Introduction

This part of ISO/IEC 14496 (MPEG-4) contains the description of all conformance bitstreams developed for the synthetic 3D graphics tools published in the following four other parts of MPEG-4: ISO/IEC 14496-11:2005, ISO/IEC 14496-16:2006, ISO/IEC 14496-21:2006 and ISO/IEC 14496-25:2009. This part of ISO/IEC 14496 also describes how tests can be designed to verify whether compressed data (i.e. bitstreams) and decoders meet the requirements specified for synthetic 3D graphics tools by those four International Standards.

Information technology — Coding of audio-visual objects —

Part 27: 3D Graphics conformance

1 Scope

This part of ISO/IEC 14496 specifies how tests can be designed to verify whether compressed data (i.e. bitstreams) and decoders meet the requirements for the synthetic 3D graphics tools specified in ISO/IEC 14496-11:2005, ISO/IEC 14496-16:2006, ISO/IEC 14496-21:2006, and ISO/IEC 14496-25:2009.

This part of ISO/IEC 14496 does not specifically address encoders. As far as synthetic 3D graphics are concerned, an encoder can be said to be an ISO/IEC 14496 encoder if it generates compressed data compliant with the syntactic and semantic bitstream payload requirements specified in ISO/IEC 14496-11, ISO/IEC 14496-16, ISO/IEC 14496-21, and ISO/IEC 14496-25.

Characteristics of coded bitstreams and decoders are defined for ISO/IEC 14496-11, ISO/IEC 14496-16, ISO/IEC 14496-21, and ISO/IEC 14496-25. The characteristics of a bitstream define the subset of the standard that is exploited in the bitstream. Examples are the applied values or range of the bitrate. Decoder characteristics define the properties and capabilities of the applied decoding process. An example of a property is the applied arithmetic accuracy. The capabilities of a decoder specify which coded bitstreams the decoder can decode and reconstruct, by defining the subset of the standard that may be exploited in decodable bitstreams. A bitstream can be decoded by a decoder if the characteristics of the coded bitstream are within the subset of the normative references.

This part of ISO/IEC 14496 describes procedures for testing conformance of compressed data and decoders to the requirements defined in ISO/IEC 14496-11, ISO/IEC 14496-16, ISO/IEC 14496-21, and ISO/IEC 14496-25; given the set of characteristics claimed, the requirements that shall be met are fully determined by these parts.

This part of ISO/IEC 14496 summarizes the requirements, cross references them to characteristics, and defines how conformance with them can be tested. Guidelines are given on constructing tests to verify decoder conformance.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14772-1:1997, *Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding*

ISO/IEC 14496-1:2004, *Information technology — Coding of audio-visual objects — Part 1: Systems*

ISO/IEC 14496-5:2001, *Information technology — Coding of audio-visual objects — Part 5: Reference software*

ISO/IEC 14496-11:2005, *Information technology — Coding of audio-visual objects — Part 11: Scene description and application engine*

ISO/IEC 14496-16:2006, *Information technology — Coding of audio-visual objects — Part 16: Animation Framework eXtension (AFX)*

ISO/IEC 14496-21:2006, *Information technology — Coding of audio-visual objects — Part 21: MPEG-J Graphics Framework eXtensions (GFX)*

ISO/IEC 14496-25:2009, *Information technology — Coding of audio-visual objects — Part 25: 3D Graphics Compression Model*

ISO/IEC 15444-1:2004, *Information technology — JPEG 2000 image coding system: Core coding system*

3 Terms definitions, abbreviations and symbols

For the purposes of this document, the terms, definitions, abbreviated terms and symbols given in ISO/IEC 14496-1, ISO/IEC 14496-5, ISO/IEC 14496-11, ISO/IEC 14496-16, ISO/IEC 14496-21, ISO/IEC 14496-25, ISO/IEC 14772-1 and ISO/IEC 15444-1 apply.

4 Tools from ISO/IEC 14496-11, Scene description and application engine, and ISO/IEC 14496-16, Animation Framework eXtension (AFX)

4.1 Scene graph nodes

4.1.1 Bitstream conformance

4.1.1.1 Conformance Requirements

BIFS streams shall comply with the specifications of Clause 8 of ISO/IEC 14496-11:2005 and Clause 4 of ISO/IEC 14496-16:2006.

4.1.1.2 Measurement procedure

The syntax of the BIFS stream shall meet the requirements of Clause 8 of ISO/IEC 14496-11:2005 and Clause 4 of ISO/IEC 14496-16:2006.

4.1.1.3 Tolerance

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

4.1.2 Terminal conformance

4.1.2.1 Conformance requirements

The terminal shall comply with the specifications of Clause 8 of ISO/IEC 14496-11:2005 and Clause 4 of ISO/IEC 14496-16:2006.

4.1.2.2 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.

4.1.2.2.1 Feature list

The test suite shall verify the features in Table 1. For nodes, the following shall be tested:

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

Table 1 — AFX test suite information

No.	Feature	Reference of test sequence and associated method
1	BitWrapper	This node shall be tested together with the AFX bitstreams in 4.2; 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	{bunny,venus}_{I,O}{C,P}{G,L}$N_1N_2N_3$, e.g., bunny_ICG101010; this node shall be tested according to 4.2.2.2.2
25	MorphSpace	cube2sphere_morph_cube, cube2sphere_morph_dome, cube2sphere_morph_etoile, cube2sphere_morph_forme, cube2sphere_morph_sphere, cube2sphere_morph_random cube2sphere_morph_anim
26	DepthImageV2	This node shall be tested together with SimpleTextureV2 nodes and PointTextureV2 node
27	SimpleTextureV2	shuttle
28	PointTextureV2	flower
29	Multitexturing	tm
30	SBVCAnimationV2	VCAnimV2
31	FootPrintSetNode	CityDynamic
32	FootPrintNode	cityFootPrintLOD
33	BuildingPartNode	CityDynamic
34	RoofNode	CityDynamic
35	FacadeNode	CityDynamic
36	Shadow	windmill, transparency

4.1.2.3 Test bitstreams

Name (for bitstream filename, add .mp4)	Provider	Content	Reference file (.wrl)
Bend	Mindego	NonLinearDeformer that bends a rectangular object.	Bend
Cyclic_GA_BIFS	VUB	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Animation of the gridCoord field.	Cyclic_MG, Cyclic_Lev0, Cyclic_Lev1, Cyclic_Lev2
Cyclic_LA_BIFS	VUB	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	VUB	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
DI_Ortho-PT_32	Samsung AIT	Orthographic projection in DepthImage node. 32 bits representation of depth value in PointTexture node.	DI_Ortho-PT_32
DI_Ortho-ST_Still	Samsung AIT	Orthographic projection in DepthImage node. Still version of SimpleTexture node.	DI_Ortho-ST_Still
DI_Ortho-ST_Anim	Samsung AIT	Orthographic projection in DepthImage node. Animated version of SimpleTexture node.	DI_Ortho- ST_Anim
DI_Persp-PT_8	Samsung AIT	Perspective projection in DepthImage node. 8 bits representation of depth value in PointTexture node.	DI_Persp-PT_8
DI_Persp-PT_32	Samsung AIT	Perspective projection in DepthImage node. 32 bits representation of depth value in PointTexture node.	DI_Persp-PT_32
DI_Persp-ST_Still	Samsung AIT	Perspective projection in DepthImage node. Still version of SimpleTexture node.	DI_Persp-ST_Still
DI_Persp-ST_Anim	Samsung AIT	Perspective projection in DepthImage node. Animated version of SimpleTexture node.	DI_Persp-ST_Anim
Humanoid_LA_BIFS	VUB	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	VUB	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
Icosa_normal	Mindego	Tagpipes sample with normal of a sector animated.	Icosa_normal
NurbsCurve	Mindego	Draw a NURBS curve.	NurbsCurve
NurbsCurve_anim	Mindego	Animate a NURBS curve.	NurbsCurve_anim
NurbsCurve2D	Mindego	Draw a 2D NURBS curve.	NurbsCurve2D
NurbsCurve2D_anim	Mindego	Animate a 2D NURBS curve.	NurbsCurve2D_anim
NurbsSurface	Mindego	Draw a NURBS surface.	NurbsSurface

Name (for bitstream filename, add .mp4)	Provider	Content	Reference file (.wrl)
OI_BVO_Still	Samsung AIT	Non-use of BitWrapper node. Non-use of voxelImageIndex. Still version of OctreeImage node.	OI_BVO_Still
OI_BVO_Anim	Samsung AIT	Non-use of BitWrapper node. Non-use of voxelImageIndex. Animated version of OctreeImage node.	OI_BVO_Anim
OI_TBVO_Still	Samsung AIT	Non-use of BitWrapper node. Use of voxelImageIndex. Still version of OctreeImage node.	OI_TBVO_Still
OI_TBVO_Anim	Samsung AIT	Non-use of BitWrapper node. Use of voxelImageIndex. Animated version of OctreeImage node.	OI_TBVO_Anim
PositionAnimator	Mindego	PositionAnimator used as an interpolator (keyType 0, keyValueType 0).	PositionAnimator
PositionAnimator_discrete	Mindego	PositionAnimator with discrete timeline (keyType 1).	PositionAnimator_discrete
PositionAnimator_linear	Mindego	PositionAnimator with linear timeline (keyType 2).	PositionAnimator_linear
PositionAnimator_NURBS_interp	Mindego	PositionAnimator with NURBS path (keyValueType 1, 2, 3) and basic interpolator (keyType 0).	PositionAnimator_NURBS_interp
PositionAnimator_paced	Mindego	PositionAnimator with paced animation (keyType 3) over a piecewise linear path (keyValueType 0).	PositionAnimator_paced
PositionAnimator_spline	Mindego	PositionAnimator with a velocity spline (keyType 4) over a piecewise linear path (keyValueType 0).	PositionAnimator_spline
PT_Default	Superscape	Procedural texture with default values.	PT_Default
PT_Gradient1	Superscape	Simple gradient - rectangle + single cell.	PT_Gradient1
PT_Gradient2	Superscape	Simple gradient - brick + 16 cells + roughness + distortion.	PT_Gradient2
PT_Gradient3	Superscape	Simple gradient - weave + 16 cells + roughness + distortion.	PT_Gradient3
PT_Gradient4	Superscape	Simple gradient - hexagonal + 16 cells + roughness + distortion.	PT_Gradient4
PT_Gradient5	Superscape	Simple gradient - ring + 4 cells + roughness.	PT_Gradient5
PT_Horizon	Superscape	Horizon texture - high roughness + low/medium distortion.	PT_Horizon
PT_Marble	Superscape	Marble texture - multiple bWeights + unequal warpmap knots.	PT_Marble
PT_PinkGranite	Superscape	Granite texture - plasma based + high roughness.	PT_PinkGranite
PT_Brickwork	Superscape	Brickwork texture - brick tiling + multiple aWeights + multiple knots + low roughness.	PT_Brickwork
PT_Fabric	Superscape	Fabric texture - weave tiling + low roughness + low distortion + multiple aWeights.	PT_Fabric
Quad_GA_BIFS	VUB	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Animation of the gridCoord field.	Quad_MG, Quad_Lev0, Quad_Lev1, Quad_Lev2
Quad_LA_BIFS	VUB	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

Name (for bitstream filename, add .mp4)	Provider	Content	Reference file (.wrl)
Quad_OA_BIFS	VUB	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
SkinnedModel	INT	Skinned and articulated model defined by using the collection of SB nodes.	SkinnedModel
Sphere_GA_BIFS	VUB	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
SS_Britney	Superscape	Extended Loop subdivision static biped model.	SS_Britney
SS_BritneyDance	Superscape	Extended Loop subdivision animated biped model.	SS_BritneyDance
SS_Goldfish	Superscape	Extended Loop subdivision textured "goldfish" model.	SS_Goldfish
Ss_img	Mindego	Shows a simple subdivision surface with 4 sectors with a texture mapped.	Ss_img
SS_RooDance	Superscape	Extended Loop subdivision animated "kangaroo" model.	SS_RooDance
SS_RooFlip	Superscape	Extended Loop subdivision animated "kangaroo" model.	SS_RooFlip
SS_Shark	Superscape	Extended Loop subdivision animated "shark" model.	SS_Shark
Tagpipes	Mindego	Two crossing cylinders as subdivision surfaces with sectors tagged.	Tagpipes
Tagpipes_anim	Mindego	Same as tagpipes sample with flatness of sectors animated.	Tagpipes_anim
Taper	Mindego	NonLinearDeformer that tapers a rectangular object.	Taper
Torus_C_LA_BIFS	VUB	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	VUB	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
cube2sphere_morph_cube	INT-ARTEMIS	MorphShape node test: static mesh obtained by morphing one base shape and four target shapes with weights 0 0 0 0.	cube2sphere_morph_cube
cube2sphere_morph_dome	INT-ARTEMIS	MorphShape node test: static mesh obtained by morphing one base shape and four target shapes with weights 0 1 0 0.	cube2sphere_morph_dome
cube2sphere_morph_etoile	INT-ARTEMIS	MorphShape node test: static mesh obtained by morphing one base shape and four target shapes with weights 0 0 1 0.	cube2sphere_morph_etoile
cube2sphere_morph_forme	INT-ARTEMIS	MorphShape node test: static mesh obtained by morphing one base shape and four target shapes with weights 0 0 0 1.	cube2sphere_morph_forme

Name (for bitstream filename, add .mp4)	Provider	Content	Reference file (.wrl)
cube2sphere_morph_sphere	INT-ARTEMIS	MorphShape node test: static mesh obtained by morphing one base shape and four target shapes with weights 1 0 0 0.	cube2sphere_morph_sphere
cube2sphere_morph_random	INT-ARTEMIS	MorphShape node test: static mesh obtained by morphing one base shape and four target shapes with weights 0.3 0.5 0.1 0.1.	cube2sphere_morph_random
cube2sphere_morph_anim	INT-ARTEMIS	MorphShape node and BBA stream test: animated mesh obtained by morphing into a morph space with one base shape and four target shapes.	cube2sphere_morph_anim
shuttle	ETH Zurich	DIBR2: SimpleTextureV2 node test: shuttle with novel fields (normal, splatU/V).	shuttle
flower	ETH Zurich	DIBR2: PointTextureV2 node test: flower with novel fields (normal, splatU/V).	flower
tm	FhG-HHI	MultiTexture and MultiTextureCoord node test: 3D Temple model with 4 Textures.	tm
VCAnimV2	INT-ARTEMIS	SBVCAnimationV2 node test: animation of a virtual character by using advanced control.	VCAnimV2
DI-PT-pos-ori-fov-plane-ortho	SAMSUNG AIT	DepthImage node with exposedFields (position, orientation, fieldofview, nearplane, farplane and orthographic) for PointTexture node.	DI-PT-pos-ori-fov-plane-ortho
DI-ST-fov-pos-plane	SAMSUNG AIT	DepthImage node with exposedFields (fieldofview, orientation, nearplane and farplane) for SimpleTexture node.	DI-ST-fov-pos-plane
DI-ST-ori-plane-fov	SAMSUNG AIT	DepthImage node with exposedFields (orientation, nearplane, farplane and fieldofview) for SimpleTexture node.	DI-ST-ori-plane-fov
cityFootPrintStatic	France Telecom	FootPrintSetNode:The building footprints of a city.	cityFootPrintStatic
cityFootPrintStaticBuffer	France Telecom	FootPrintSetNode: The building footprints of a city.	cityFootPrintStaticBuffer
cityFootPrintLOD	France Telecom	FootPrintSetNode: The building footprints of a city with LOD.	cityFootPrintLOD
cityFootPrintLODBuffer	France Telecom	FootPrintSetNode:The building footprints of a city with LOD.	cityFootPrintLODBuffer
cityFootPrintLOD_0_001	France Telecom	FootPrintSetNode:The building footprints of a city with LOD using the compression based on the scene accuracy.	cityFootPrintLOD_0_001
cityFootPrintLOD_0_001Buffer	France Telecom	FootPrintSetNode:The building footprints of a city with LOD using the compression based on the scene accuracy.	cityFootPrintLOD_0_001Buffer
cityStatic	France Telecom	FootPrintSetNode:The 3D building of a city.	cityStatic
cityStaticBuffer	France Telecom	FootPrintSetNode:The 3D building of a city.	cityStaticBuffer
cityDynamic	France Telecom	FootPrintSetNode:The 3D building of a city with LOD using the compression based on the scene accuracy.	cityDynamic

Name (for bitstream filename, add .mp4)	Provider	Content	Reference file (.wrl)
cityDynamicBuffer	France Telecom	FootPrintSetNode:The 3D building of a city with LOD using the compression based on the scene accuracy.	cityDynamicBuffer
house	France Telecom	FootPrintSetNode:The complex 3D model of a house.	house
windmill	University of Ilmenau	Shadow: Animated windmill, that casts shadows on the ground.	windmill
transparent	University of Ilmenau	Shadow: Several cubes with different transparency cast shadows on the ground.	transparent

4.1.2.4 Tolerance

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

4.1.3 Rendering conformance

4.1.3.1 Conformance requirements

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

4.1.3.2 Measurement procedure and tolerance

4.1.3.2.1 ProceduralTexture

Nodes: ProceduralTexture

Provider: Superscape

Bitstreams: 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.

4.2 Elementary bitstreams

4.2.1 Common conformance point

4.2.1.1 Bitstream conformance

4.2.1.1.1 Conformance requirements

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

4.2.1.1.2 Measurement procedure

The syntax of the AFX bitstreams shall meet the requirements specified in 7.2.6 of ISO/IEC 14496-1:2004.

4.2.1.1.3 Tolerance

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

4.2.2 Geometry bitstreams

4.2.2.1 3DMC extension

The 3D mesh object is a 3D polygonal model that can be represented as an IndexedFaceSet in BIFS. It is defined by the position of its vertices (geometry), by the association between each face and its sustaining vertices (connectivity), and optionally by colours, normals, and texture coordinates (properties). Properties do not affect the 3D geometry, but influence the way the model is shaded.

3D mesh coding (3DMC) extension addresses the efficient coding of 3D mesh object. It comprises a basic method and several options. The basic 3DMC extension method operates on manifold model and features incremental representation of single resolution 3D model. The model may be triangular or polygonal – the latter are triangulated for coding purposes and are fully recovered in the decoder. Options include: (a) support for error resilience; (b) vertex order and face order preserving; (c) efficient texture mapping; and (d) support for non-manifold and non-orientable model.

4.2.2.1.1 Conformance points

4.2.2.1.1.1 Covered functionalities

The conformance points for compression of 3DMC extension cover basic compression, backward compatibility, forward compatibility, error resilience support, backward compatibility with error resilience, forward compatibility with error resilience, vertex order and face order preserving, support of non-manifold/non-orientable model (stitch), and efficient texture mapping. These functionalities relate to the compressed representation of the IndexedFaceSet node carried by the BitWrapper node as described in 7.2.2.23 of ISO/IEC 14496-11:2005.

As for carriage of compressed representation by the IndexedFaceSet node using the BitWrapper node, it can be carried either in a separate stream or within the scene stream (BIFS stream). Therefore, compression of 3DMC extension shall also be tested together with this node.

The following Subclauses specify the normative tests for verifying conformance of 3DMC extension compressed bitstreams and 3DMC extension decoder. Those normative tests make use of test data (bitstream test suites).

4.2.2.1.2 Bitstream conformance

4.2.2.1.2.1 Conformance requirements

BIFS streams shall comply with the specifications for compression of 3DMC extension in ISO/IEC 14496-16:2006/Amd1:2007 and 7.2.2.23 of ISO/IEC 14496-11:2005.

4.2.2.1.2.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of compression of 3DMC extension in ISO/IEC 14496-16:2006/Amd1:2007 and 7.2.2.23 of ISO/IEC 14496-11:2005.

4.2.2.1.2.3 Tolerance

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

4.2.2.1.3 Terminal conformance

4.2.2.1.3.1 Conformance requirements

A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-16:2006/Amd1:2007 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 14496-11:2005 shall also be implemented.

4.2.2.1.3.2 Test bitstreams

Purpose 1 (URL): Exercise the basic compression, backward compatibility, forward compatibility, error resilience support, backward compatibility with error resilience, forward compatibility with error resilience, vertex order and face order preserving, support of non-manifold/non-orientable model (stitch), and efficient texture mapping functionalities of MPEG-4 3DMC extension compression carried in a separate stream from the scene stream.

Purpose 2 (Buffer): Exercise the basic compression, backward compatibility, forward compatibility, error resilience support, backward compatibility with error resilience, forward compatibility with error resilience, vertex order and face order preserving, support of non-manifold/non-orientable model (stitch), and efficient texture mapping functionalities of MPEG-4 3DMC extension compression carried in a BIFS scene stream.

Files:

Name	**Attribute**	**Bitstream (.mp4)**	**Reference file (.wrl)**
BASIC#1-1	Use of URL field in the Bitwrapper node. Use of basic compression of polygonal 3D mesh with geometry, connectivity, color, normal, and texture coordinates.	BASIC-object #1-1	BASIC-object#1-1
BASIC#1-2	Use of URL field in the Bitwrapper node. Use of basic compression of triangle 3D mesh with geometry, connectivity, color, normal, and texture coordinates composed of 1 connected component.	BASIC-object #1-2	BASIC-object#1-2
BASIC#1-3	Use of URL field in the Bitwrapper node. Use of basic compression of triangle 3D mesh with geometry, connectivity, color, normal, and texture coordinates composed of several connected components.	BASIC-object #1-3	BASIC-object#1-3
CROSS#1-1	Use of URL field in the Bitwrapper node. Use of backward compatibility. Use of 3DMC decoder.	CROSS-object #1-1	CROSS-object#1-1
CROSS#1-2	Use of URL field in the Bitwrapper node. Use of forward compatibility.	CROSS-object#1-2	CROSS-object#1-2
ERR_MODE #1-1	Use of URL field in the Bitwrapper node. Use of error resilience mode (error resilience mode: 0, packetsize: 360, boundary predicition type: 0)	ERR_MODE-object#1-1	ERR_MODE-object#1-1
ERR_MODE #1-2	Use of URL field in the Bitwrapper node. Use of error resilience mode (error resilience mode: 0, packetsize: 360, boundary predicition type: 1)	ERR_MODE-object#1-2	ERR_MODE-object#1-2
ERR_MODE #1-3	Use of URL field in the Bitwrapper node. Use of error resilience mode (error resilience mode: 1, packetsize: 360, boundary predicition type: 0)	ERR_MODE-object#1-3	ERR_MODE-object#1-3
ERR_MODE #1-4	Use of URL field in the Bitwrapper node. Use of error resilience mode (error resilience mode: 1, packetsize: 360, boundary predicition type: 1)	ERR_MODE-object#1-4	ERR_MODE-object#1-4

Name	Attribute	Bitstream (.mp4)	Reference file (.wrl)
ERR_CROSS#1-1	Use of URL field in the Bitwrapper node. Use of backward compatibility and error resilience mode (error resilience mode: 0, packetsize: 360, boundary predicition type: 0). Use of 3DMC decoder.	ERR_CROSS-object#1-1	ERR_CROSS -object#1-1
ERR_CROSS#1-2	Use of URL field in the Bitwrapper node. Use of forward compatibility and error resilience mode (error resilience mode: 0, packetsize: 360, boundary predicition type: 0)	ERR_CROSS -object#1-2	ERR_CROSS -object#1-2
ORDER#1-1	Use of URL field in the Bitwrapper node. Use of vertex order preserving coded at the unit of IFS	ORDER-object#1-1	ORDER -object#1-1
ORDER#1-2	Use of URL field in the Bitwrapper node. Use of vertex order preserving coded at the unit of CC	ORDER -object#1-2	ORDER -object#1-2
ORDER#1-3	Use of URL field in the Bitwrapper node. Use of face order preserving coded at the unit of IFS	ORDER -object#1-3	ORDER -object#1-3
ORDER#1-4	Use of URL field in the Bitwrapper node. Use of face order preserving coded at the unit of CC	ORDER -object#1-4	ORDER -object#1-4
ORDER#1-5	Use of URL field in the Bitwrapper node. Use of vertex order and face order preserving coded at the unit of IFS	ORDER -object#1-5	ORDER -object#1-5
ORDER#1-6	Use of URL field in the Bitwrapper node. Use of vertex order and face order preserving coded at the unit of CC	ORDER -object#1-6	ORDER -object#1-6
STITCH#1-1	Use of URL field in the Bitwrapper node. Use of support of non-manifold model.	STITCH-object#1-1	STITCH -object#1-1
STITCH#1-2	Use of URL field in the Bitwrapper node. Use of support of non-manifold and non-orientable model.	STITCH -object#1-2	STITCH -object#1-2
TC#1-1	Use of URL field in the Bitwrapper node. Use of efficient texture mapping with use of Adaptive_quant_texCoord_mode.	TC-object#1-1	TC-object#1-1
TC#1-2	Use of URL field in the Bitwrapper node. Use of efficient texture mapping with no use of Adaptive_quant_texCoord_mode.	TC-object#1-2	TC-object#1-2
BASIC#2-1	Use of buffer field in the Bitwrapper node. Use of basic compression of polygonal 3D mesh with geometry, connectivity, color, normal, and texture coordinates.	BASIC-object #2-1	BASIC-object#2-1
BASIC#2-2	Use of buffer field in the Bitwrapper node. Use of basic compression of triangle 3D mesh with geometry, connectivity, color, normal, and texture coordinates composed of 1 connected component.	BASIC-object #2-2	BASIC-object#2-2
BASIC#2-3	Use of buffer field in the Bitwrapper node. Use of basic compression of triangle 3D mesh with geometry, connectivity, color, normal, and texture coordinates composed of several connected components.	BASIC-object #2-3	BASIC-object#2-3
CROSS#2-1	Use of buffer field in the Bitwrapper node. Use of backward compatibility. Use of 3DMC decoder.	CROSS-object #2-1	CROSS-object#2-1
CROSS#2-2	Use of buffer field in the Bitwrapper node. Use of forward compatibility.	CROSS -object #2-2	CROSS -object#2-2
ERR_MODE#2-1	Use of buffer field in the Bitwrapper node. Use of error resilience mode (error resilience mode: 0, packetsize: 360, boundary predicition type: 0)	ERR_MODE-object#2-1	ERR_MODE -object#2-1

Name	Attribute	Bitstream (.mp4)	Reference file (.wrl)
ERR_MODE #2-2	Use of buffer field in the Bitwrapper node. Use of error resilience mode (error resilience mode: 0, packetsize: 360, boundary predicition type: 1)	ERR_MODE - object#2-2	ERR_MODE - object#2-2
ERR_MODE #2-3	Use of buffer field in the Bitwrapper node. Use of error resilience mode (error resilience mode: 1, packetsize: 360, boundary predicition type: 0)	ERR_MODE - object#2-3	ERR_MODE - object#2-3
ERR_MODE #2-4	Use of buffer field in the Bitwrapper node. Use of error resilience mode (error resilience mode: 1, packetsize: 360, boundary predicition type: 1)	ERR_MODE - object#2-4	ERR_MODE - object#2-4
ERR_CROSS#2-1	Use of buffer field in the Bitwrapper node. Use of backward compatibility and error resilience mode (error resilience mode: 0, packetsize: 360, boundary predicition type: 0). Use of 3DMC decoder.	ERR_CROSS-object#2-1	ERR_CROSS - object#2-1
ERR_CROSS#2-2	Use of buffer field in the Bitwrapper node.Use of forward compatibility and error resilience mode (error resilience mode: 0, packetsize: 360, boundary predicition type: 0)	ERR_CROSS - object#2-2	ERR_CROSS - object#2-2
ORDER#2-1	Use of buffer field in the Bitwrapper node. Use of vertex order preserving coded at the unit of IFS	ORDER-object#2-1	ORDER -object#2-1
ORDER#2-2	Use of buffer field in the Bitwrapper node. Use of vertex order preserving coded at the unit of CC	ORDER -object#2-2	ORDER -object#2-2
ORDER#2-3	Use of buffer field in the Bitwrapper node. Use of face order preserving coded at the unit of IFS	ORDER -object#2-3	ORDER -object#2-3
ORDER#2-4	Use of buffer field in the Bitwrapper node. Use of face order preserving coded at the unit of CC	ORDER -object#2-4	ORDER -object#2-4
ORDER#2-5	Use of buffer field in the Bitwrapper node. Use of vertex order and face order preserving coded at the unit of IFS	ORDER -object#2-5	ORDER -object#2-5
ORDER#2-6	Use of buffer field in the Bitwrapper node. Use of vertex order and face order preserving coded at the unit of CC	ORDER -object#2-6	ORDER -object#2-6
STITCH#2-1	Use of buffer field in the Bitwrapper node. Use of support of non-manifold model.	STITCH-object#2-1	STITCH -object#2-1
STITCH#2-2	Use of buffer field in the Bitwrapper node. Use of support of non-manifold and non-orientable model.	STITCH -object#2-2	STITCH -object#2-2
TC#2-1	Use of buffer field in the Bitwrapper node. Use of efficient texture mapping with use of Adaptive_quant_texCoord_mode.	TC-object#2-1	TC-object#2-1
TC#2-2	Use of buffer field in the Bitwrapper node. Use of efficient texture mapping with no use of Adaptive_quant_texCoord_mode.	TC-object#2-2	TC-object#2-2

4.2.2.1.3.3 Measurement procedure

The terminal should produce a formatted output giving the reconstructed fields of IndexedFaceSet. The decoder shall be able to decode the bitstreams provided as described in 4.2.2.1.3.2.

4.2.2.1.3.4 Tolerance

The diagnosis is to check whether the field data (geometry, connectivity, and properties) of IndexedFaceSet node that is decoded from ".mp4" files correspond with the node included in the provided reference ".wrl" files.

4.2.2.2 Wavelet subdivision surfaces

4.2.2.2.1 Bitstream conformance

4.2.2.2.1.1 Conformance requirements

BIFS streams shall comply with the specifications in 5.1 of ISO/IEC 14496-16:2006.

4.2.2.2.1.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of 5.1 of ISO/IEC 14496-16:2006.

4.2.2.2.1.3 Tolerance

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

4.2.2.2.2 Terminal conformance

4.2.2.2.2.1 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.

4.2.2.2.2.2 Test bitstreams

Model name	Provider	In/Out-band	Complete/ Partial	Global/Local	Quantisation	Bitstream (.mp4)
bunny	UPM	I	C	G (x, y, z)	(10, 10, 10)	bunny_ICG101010
					(10, 09, 08)	bunny_ICG100908
				L (n, t_1, t_2)	(10, 10, 10)	bunny_ICL101010
					(10, 09, 08)	bunny_ICL100908
venus	UPM	O	C	G (x, y, z)	(10, 10, 10)	venus_OCG101010
					(10, 09, 08)	venus_OCG100908
				L (n, t_1, t_2)	(10, 10, 10)	venus_OCL101010
					(10, 09, 08)	venus_OCL100908
bunny	France Telecom R&D	I	P	G (x, y, z)	(10, 10, 10)	bunny_IPG101010
					(10, 09, 08)	bunny_IPG100908
				L (n, t_1, t_2)	(10, 10, 10)	bunny_IPL101010
					(10, 09, 08)	bunny_IPL100908
venus	France Telecom R&D	O	P	G (x, y, z)	(10, 10, 10)	venus_OPG101010
					(10, 09, 08)	venus_OPG100908
				L (n, t_1, t_2)	(10, 10, 10)	venus_OPL101010
					(10, 09, 08)	venus_OPL100908

The name of all bitstreams follow the pattern "*modelName_*$A_1A_2A_3N_1N_2N_3$.mp4", where:

- *modelName* is either "bunny" or "venus", the corresponding 3D model being, respectively, the well-known bunny from the Stanford 3D scanning repository or the also classical venus head;
- A_1 is either "I" or "O", standing respectively for In-band or Out-band stream;
- A_2 is either "C" or "P", indicating respectively an either Complete or Partial spatial coding of the wavelet coefficient tree forest;
- A_3 is either "G" or "L", meaning respectively that the coordinate frame for the 3D details is either Global (and hence the coordinates are x, y and z) or Local (and hence the coordinates are n, t_1 and t_2);
- N_1, N_2 and N_3 are integers between 8 and 10 (printed with two digits) reflecting the number of bits devoted to the quantisation of the first, second and third coordinate, respectively.

For instance, "bunny_ICG100908.mp4" is the WaveletSubdivisionSurface representation of the bunny model in an in-band stream with complete spatial encoding, global coordinates, and different quantisation for different components (10 for x, 9 for y and 8 for z).

4.2.2.2.2.3 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.

4.2.2.2.2.4 Tolerance

The conformance is passed when the reconstructed mesh reflects the correct mesh, as defined in ISO/IEC 14496-16:2006. The rendering functionality must be observed visually.

4.2.2.3 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 4.3.4 of ISO/IEC 14496-16:2006. The fields of the MeshGrid node are encoded either by BIFS, or in a compressed representation as a binary stream as explained in 5.2 of ISO/IEC 14496-16:2006. In both cases the information needs to be decoded to obtain the model as given in 5.2.3 of ISO/IEC 14496-16:2006. 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 4.3.4.3 of ISO/IEC 14496-16:2006.

4.2.2.3.1 Conformance points

4.2.2.3.1.1 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 the BitWrapper node as described in 7.2.2.23 of ISO/IEC 14496-11:2005. 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 shall also be tested together with this node as described in ISO/IEC 14496-11:2005.

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/Amd7:2005 with source code available in electronic format.

4.2.2.3.2 Bitstream conformance

4.2.2.3.2.1 Conformance requirements

BIFS streams shall comply with the specifications in 5.2 of ISO/IEC 14496-16:2006.

4.2.2.3.2.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of 5.2 of ISO/IEC 14496-16:2006.

4.2.2.3.2.3 Tolerance

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

4.2.2.3.3 Terminal conformance

4.2.2.3.3.1 Conformance requirements

The terminal shall decode the BIFS stream and initialize the fields of the object of 4.3.4 of ISO/IEC 14496-16:2006.

4.2.2.3.3.2 Test bitstreams

4.2.2.3.3.2.1 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 4.3.4.2 of ISO/IEC 14496-16:2006.

Name (for bitstream filename, add .mp4)	Provider	Description	Reference file (.xml)
Torus_C_LA_BIFS	VUB	Multi-resolution non-homogeneous mesh with uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field.	Torus_BIFS
Torus_C_OA_BIFS	VUB	Multi-resolution non-homogeneous mesh with uniform distributed reference grid. Animation of the vertexOffset field.	Torus_BIFS
Humanoid_LA_BIFS	VUB	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field.	Humanoid_BIFS
Humanoid_OA_BIFS	VUB	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid. Animation of the vertexOffset field.	Humanoid_BIFS
Sphere_GA_BIFS	VUB	Multi-resolution non-homogeneous mesh. Animation of the gridCoord field.	Sphere_BIFS
Quad_LA_BIFS	VUB	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field.	Quad_BIFS
Quad_OA_BIFS	VUB	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Animation of the vertexOffset field.	Quad_BIFS

Name (for bitstream filename, add .mp4)	Provider	Description	Reference file (.xml)
Quad_GA_BIFS	VUB	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Animation of the gridCoord field.	Quad_BIFS
Cyclic_LA_BIFS	VUB	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field.	Cyclic_BIFS
Cyclic_OA_BIFS	VUB	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Animation of the vertexOffset field.	Cyclic_BIFS
Cyclic_GA_BIFS	VUB	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Animation of the gridCoord field.	Cyclic_BIFS

4.2.2.3.3.2.2 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 4.2.2.3.3.2.1.

Name (for bitstream filename, add .mp4)	Provider	Description	Reference file (.xml)
Torus_C_URL	VUB	Multi-resolution non-homogeneous mesh with uniform distributed reference grid.	Torus_MG.
Humanoid_URL	VUB	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid.	Humanoid_MG
Quad_URL	VUB	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1).	Quad_MG
Cyclic_URL	VUB	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1).	Cyclic_MG

4.2.2.3.3.2.3 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 4.2.2.3.3.2.1.

Name (for bitstream filename, add .mp4)	Provider	Description	Reference file (.xml)
Torus_C_LA_BUFFER	VUB	Multi-resolution non-homogeneous mesh with uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field	Torus_MG
Humanoid_LA_BUFFER	VUB	Multi-resolution non-homogeneous mesh with non-uniform distributed reference grid. Different resolution levels are displayed by animating the displayLevel field	Humanoid_MG
Quad_LA_BUFFER	VUB	Multi-resolution homogeneous quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field	Quad_MG
Cyclic_LA_BUFFER	VUB	Multi-resolution cyclic quadrilateral mesh (uniformSplit = 1). Different resolution levels are displayed by animating the displayLevel field	Cyclic_MG

4.2.2.3.3.3 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 4.2.2.3.3.2. The decoder shall be able to decode the bitstreams provided as described in 4.2.2.3.3.2. The output should include the description of the following fields of the node description defined in 4.3.4 of ISO/IEC 14496-16:2006:

- 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.

4.2.2.3.3.4 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 4.3.4.1.1 of ISO/IEC 14496-16:2006.

The *ConnectivityWireframe* section lists, for each resolution level, the vertices in the order they are decoded and their attributes as defined in the Subclause from below of ISO/IEC 14496-16:2006: (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*) (Subclauses 4.3.4.3.2 and 5.2.3.3), (4) the border direction (*borderDir*) (Subclauses 4.3.4.3.2 and 5.2.3.1), (5) the coordinates (*x*, *y*, *z*) (Subclauses 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.

4.2.2.3.3.5 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/scale$_{(x,y,z)}$ compared to the sample values, where scale$_{(x,y,z)}$ are the scaling factors (*gridScale*) defined for each of the x, y, z grid coeficients as specified in 5.2.2.2 of ISO/IEC 14496-16:2006, (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.

4.2.2.4 Multiresolution footprint

4.2.2.4.1 Introduction

Some 3D models such as buildings can be well-compressed using a footprint-based representation. To allow an adaptive streaming of huge models, this coding is based on a multi-resolution schema well-suited for graphics on-demand transmission. For the specific case of building models, a procedural representation is available to encode complex facades and roofs. Due to its compression, the multi-resolution footprint-based coding is actually one of the best way to transmit huge city models for geo-visualization applications.

4.2.2.4.2 Conformance points

4.2.2.4.2.1 Covered functionalities

The conformance points for the multiresolution footprint based representation cover the different object type (0=footprint, 1=building), the static (with one level of detail) or dynamic scene (with several levels of detail), the compression based on the accuracy of the scene, the different fields use in the Bitwrapper node (url or BUFFER), the swap nodes used in building mode (allowing to replace the encoded building by a more complex indexedFaceSet), the facades reconstruction with textures or 3D models, the complex roofs reconstruction with simple roof superimposition.

The following Subclauses specify the normative tests for verifying conformance of the multiresolution footprint based representation. Those normative tests make use of test data (bitstream test suites).

4.2.2.4.3 Bitstream conformance

4.2.2.4.3.1 Conformance requirements

BIFS streams shall comply with the specifications in 4.3.6 of ISO/IEC 14496-16:2006 and 7.2.2.23 of ISO/IEC 14496-11:2005.

4.2.2.4.3.2 Measurement procedure

BIFS streams shall comply with the specifications in 4.3.6 of ISO/IEC 14496-16:2006 and 7.2.2.23 of ISO/IEC 14496-11:2005.

4.2.2.4.3.3 Tolerance

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

4.2.2.4.4 Terminal conformance

4.2.2.4.4.1 Conformance requirements

A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-16:2006 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 14496-11:2005 shall also be implemented.

4.2.2.4.4.2 Test bitstreams

Name (for bitstream filename, add .mp4)	Attribute	Reference file (.wrl)
Standalone_URL	Use of URL field in the BitWrapper node. Entire object coded at once.	Standalone_URL
Standalone_Buffer	Use of URL field in the BitWrapper node. Entire object coded at once.	Standalone_Buffer
Partial_URL	Use of URL in the BitWrapper node. Partially encoded object.	Partial_URL
Partial_BUFFER	Use of BUFFER in the BitWrapper node. Partially encoded object.	Partial_BUFFER
Progressive_URL	Use of URL in the BitWrapper node. Refinement to Partial_URL.	Progressive_URL
Progressive_BUFFER	Use of BUFFER in the BitWrapper node. Refinement to Partial_Buffer.	Progressive_BUFFER

4.2.2.4.4.3 Measurement procedure

The terminal should produce a formatted output giving the reconstructed 3D object with the available refinement Elementary Streams.

4.2.2.4.4.4 Tolerance

The conformance is passed when the reconstructed 3D object reflects the correct 3D object, as defined in ISO/IEC 14496-16:2006. The rendering functionality must be observed visually.

4.2.3 Appearance bitstreams

4.2.3.1 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).

4.2.3.1.1 Conformance points

4.2.3.1.1.1 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 7.2.2.23 of ISO/IEC 14496-11:2005.

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 7.2.2.23 of ISO/IEC 14496-11:2005.

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/Amd7:2005 with source code available in electronic format.

4.2.3.1.2 Bitstream conformance

4.2.3.1.2.1 Conformance requirements

BIFS streams shall comply with the specifications of Clause 8 of ISO/IEC 14496-11:2005.

4.2.3.1.2.2 Measurement procedure

The syntax of the BIFS stream shall meet the requirements of Clause 8 of ISO/IEC 14496-11:2005.

4.2.3.1.2.3 Tolerance

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

4.2.3.1.3 Terminal conformance

4.2.3.1.3.1 Conformance requirements

A decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-16:2006 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 14496-11:2005 shall also be implemented.

4.2.3.1.3.2 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 4.2.3.1.3.4.

4.2.3.1.3.3 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.

4.2.3.1.3.4 Test bitstreams

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	OI_Bitwrapper-URL _TBVO_Still	OI_Bitwrapper-URL _TBVO_Still

	version of OctreeImage node.		
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

4.2.3.2 JPEG 2000 support

4.2.3.2.1 Introduction

Initially, the only still image formats natively supported by the MPEG-4 specification, other than its own, VTC (Visual Texture Coding), were JPEG and PNG. More precisely, only ESs (Elementary Streams) describing still images according to those formats could be inserted in an MPEG-4 bitstream in the natural manner, i.e., by multiplexing them with the other ESs of the compound stream and having an ImageTexture node (from the BIFS-coded ES of the scene) refer to them. It was impossible to do this for textures coded according to the JPEG 2000 format, which provides in many cases great advantages over both JPEG and PNG, both in terms of compression efficiency and multi-resolution handles.

4.2.3.2.2 Conformance points

4.2.3.2.2.1 Covered functionalities

The single conformance point of the JPEG 2000 support is to enable the use of ESs (Elementary Streams) containing textures coded according to the JPEG 2000 standard within MPEG-4 bitstreams.

4.2.3.2.3 Bitstream conformance

4.2.3.2.3.1 Conformance requirements

MPEG-4 BIFS streams shall comply with the specifications in ISO/IEC 14496-1:2004. JPEG 2000 ESs shall comply with the specifications in ISO/IEC 15444-1:2004.

4.2.3.2.3.2 Measurement procedure

MPEG-4 BIFS streams shall comply with the specifications in ISO/IEC 14496-1:2004. JPEG 2000 ESs shall comply with the specifications in ISO/IEC 15444-1:2004.

4.2.3.2.3.3 Tolerance

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

4.2.3.2.4 Terminal conformance

4.2.3.2.4.1 Conformance requirements

A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-1:2004, as far as the decoding of the scene is concerned, and to the one specified in ISO/IEC 15444-1:2004, as far as the texture is concerned. The decoder shall decode bitstreams with any options or parameters with values permitted for the considered functionalities.

4.2.3.2.4.2 Test bitstreams

Name	Attribute	Bitstream (.mp4)	Reference files (.{wrl,scr})
Quad	Simple use of a JPEG 2000 ES inside an MPEG-4 bitstream, with no hints for the decoder. The scene consists of a JPEG 2000 texture mapped onto a rectangle defined by two triangles.	quad	quad
Quad with DSI (Decoder Specific Information)	Exactly as above, but with hints for the decoder, which are given through decSpecificInfo JPEG2000_DecoderConfig {} within decConfigDescr DecoderConfigDescriptor {} of the ES_Descriptor {} of the ES containing the texture.	quad_dsi	quad_dsi
Earth	Slightly more complex use of a JPEG 2000 ES inside an MPEG-4 bitstream, with no hints for the decoder. The scene consists of a JPEG 2000 texture mapped onto a sphere tesselated by many triangles.	earth	earth
Earth with DSI	Exactly as above, but with hints for the decoder, which are given through decSpecificInfo JPEG2000_DecoderConfig {} within decConfigDescr DecoderConfigDescriptor {} of the ES_Descriptor {} of the ES containing the texture.	earth_dsi	earth_dsi

4.2.3.2.4.3 Measurement procedure

The terminal should produce a formatted output showing any textures contained in JPEG 2000 ESs within the MPEG-4 bitstream correctly decoded.

4.2.3.2.4.4 Tolerance

The terminal conformance is passed if the textures contained in JPEG 2000 ESs within the MPEG-4 bitstream are correctly rendered, which must be observed visually.

4.2.3.3 PointTexture

PointTexture node in ISO/IEC 14496-16:2006 has the depth information and the color information. PointTexture represents an object with an array of pixels viewed from a single camera location. Each PointTexture pixel is represented by its color, depth (the distance from the pixel to the camera), and a few other properties assisting PointTexture rendering. There can be multiple pixels along each line of sight, and thus a PointTexture usually consists of multiple layers. A PointTexture typically contains a large amount of data: a realistic image requires a higher sampling density and a huge amount of data. Therefore, the compression of PointTexture images should be done efficiently.

The PointTexture compression is a tool to compress the PointTexture node efficiently.

4.2.3.3.1 Conformance points

4.2.3.3.1.1 Covered functionalities

The conformance points for PointTexture compression covers lossless compression and lossy compression. These functionalities relate to the compressed representation of PointTexture node carried by BitWrapper node as described in 7.2.2.23 of ISO/IEC 14496-11:2005.

As for carriage of compressed representation of PointTexture node using BitWrapper node, it can be carried either in a separate stream or within the scene stream (BIFS stream). Therefore, PointTexture compression shall also be tested together with this node.

The following Subclauses specify the normative tests for verifying the conformance of PointTexture compressed bitstreams and PointTexture decoder. Those normative tests make use of test data (bitstream test suites).

4.2.3.3.2 Bitstream conformance

4.2.3.3.2.1 Conformance requirements

BIFS streams shall comply with the specifications for PointTexture compression in ISO/IEC 14496-16:2006 and 7.2.2.23 of ISO/IEC 14496-11:2005.

4.2.3.3.2.2 Measurement procedure

The syntax of the BIFS stream shall meet the requirements of PointTexture compression in ISO/IEC 14496-16:2006 and 7.2.2.23 of ISO/IEC 14496-11:2005.

4.2.3.3.2.3 Tolerance

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

4.2.3.3.3 Terminal conformance

4.2.3.3.3.1 Conformance requirements

A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-16:2006 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 14496-11:2005 shall also be implemented.

4.2.3.3.3.2 Test bitstreams

Purpose 1 (URL): Exercise the lossless and lossy compression functionality of MPEG-4 PointTexture compression carried in a separate stream from the scene stream.

Purpose 2 (Buffer): Exercise the lossless and lossy compression functionality of MPEG-4 PointTexture compression carried in the scene stream.

Files:

Name	Attribute	Bitstream (.mp4)	Reference file (.wrl - DIBR only)
PT#1-1	Use of URL field in the Bitwrapper node. Use of lossless compression (nPercentOfDecoding is set to 100)	PT-object #1-1	PT-object#1-1
PT#1-2	Use of URL field in the Bitwrapper node. Use of lossy compression (nPercentOfDecoding is set to 99)	PT-object #1-2	PT-object#1-2
PT#2-1	Use of buffer field in the Bitwrapper node. Use of lossless compression (nPercentOfDecoding is set to 100)	PT-object #2-1	PT-object#2-1
PT#2-2	Use of buffer field in the Bitwrapper node. Use of lossy compression (nPercentOfDecoding is set to 99)	PT-object #2-2	PT-object#2-2

4.2.3.3.3.3 Measurement procedure

The terminal should produce a formatted output giving the reconstructed fields of PointTexture. The decoder shall be able to decode the bitstreams provided as described in 4.2.3.3.3.2.

4.2.3.3.3.4 Tolerance

The diagnosis is to check whether the field data (depth, color, width, height, depthNbBits) of PointTexture node that is decoded from “.mp4” files correspond with the node included in the provided reference “.wrl” files.

4.2.4 Animation bitstreams

4.2.4.1 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 14496-11:2005. The functionalities are:

1. Key preserving (compress interpolators while preserving key and keyValue structure for random access): coordinate interpolator, orientation interpolator, position interpolator.

2. Path preserving (compress interpolators without necessarily preserving all keys and key values for maintaining animation path): orientation interpolator, position interpolator.

4.2.4.1.1 Conformance points

4.2.4.1.1.1 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 the BitWrapper node as described in 7.2.2.23 of ISO/IEC 14496-11:2005.

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 shall also be tested together with this node as described in 7.2.2.23 of ISO/IEC 14496-11:2005.

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/Amd7:2005 with source code available in electronic format.

4.2.4.1.2 Bitstream conformance

4.2.4.1.2.1 Conformance requirements

BIFS streams shall comply with the specifications of Clause 8 of ISO/IEC 14496-11:2005.

4.2.4.1.2.2 Measurement procedure

The syntax of the BIFS stream shall meet the requirements of Clause 8 of ISO/IEC 14496-11:2005.

4.2.4.1.2.3 Tolerance

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

4.2.4.1.3 Terminal conformance

4.2.4.1.3.1 Coordinate interpolator

4.2.4.1.3.1.1 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 14496-11:2005 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 14496-11:2005 shall also be implemented.

In the following Subclauses the term ‘reference decoder’ means the technical report software verifier (ISO/IEC 14496-5:2001/Amd7:2005). The reference decoder is a decoder that implements precisely the decoding process as specified in ISO/IEC 14496-11:2005.

4.2.4.1.3.1.2 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 4.2.4.1.3.1.3.

4.2.4.1.3.1.3 Test bitstreams for key preserving

4.2.4.1.3.1.3.1 Test bitstreams key preserving #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.

Files:

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

Name	Attribute	Bitstream info. for key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
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

4.2.4.1.3.1.3.2 Test bitstreams key preserving #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.

Files:

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

Name	Attribute	Bitstream info. for key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
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 the 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 the 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 the 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) appears in the dictionary position indices	nTrueOne‘s are all 0’s	ci_kp#2-ZeroTrueOne	ci_kp#2-ZeroTrueOne

4.2.4.1.3.1.3.3 Test bitstreams key preserving #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.

Files:

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

4.2.4.1.3.1.3.4 Test bitstreams key preserving #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.

Files:

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

Name	Attribute	Bitstream info. for key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
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	test selection flag when some quantization values are the 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 the 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

Name	Attribute	Bitstream info. for key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
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 the 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) appears in the dictionary position indices	nTrueOne‘s are all 0’s	ci_kp#4-ZeroTrueOne	ci_kp#4-ZeroTrueOne

4.2.4.1.3.1.4 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.

4.2.4.1.3.2 Orientation interpolator

4.2.4.1.3.2.1 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 14496-11:2005 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 14496-11:2005 shall also be implemented.

In the following Subclauses the term ‘reference decoder’ means the technical report software verifier (ISO/IEC 14496-5:2001/Amd7:2005). The reference decoder is a decoder that implements precisely the decoding process as specified in ISO/IEC 14496-11:2005.

4.2.4.1.3.2.2 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 4.2.4.1.3.2.3 and 4.2.4.1.3.2.4.

4.2.4.1.3.2.3 Test bitstreams for key preserving

4.2.4.1.3.2.3.1 Test bitstreams key preserving #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.

Files:

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

4.2.4.1.3.2.3.2 Test bitstreams key preserving #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.

Files:

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

4.2.4.1.3.2.3.3 Test bitstreams key preserving #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.

Files:

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

Name	Attribute	Bitstream info. for key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
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

4.2.4.1.3.2.3.4 Test bitstreams key preserving #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.

Files:

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

4.2.4.1.3.2.4 Test bitstreams for path preserving

4.2.4.1.3.2.4.1 Test bitstreams path preserving #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.

Files:

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	bKeyInvertDownFlag is enabled	oi_pp#1-KeyInvertDown	oi_pp#1-KeyInvertDown
oi_pp#1-13	SignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is 1~7	oi_pp#1-NoKeyInvertDownAndSignedAAC	oi_pp#1-NoKeyInvertDownAndSignedAAC
oi_pp#1-14	UnsignedAAC decoding	bKeyInvertDownFlag is disabled when nDNDOrder is –1~0	oi_pp#1-NoKeyInvertDownAndUnsignedAAC	oi_pp#1-NoKeyInvertDownAndUnsignedAAC

4.2.4.1.3.2.4.2 Test bitstreams path preserving #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.

Files:

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

4.2.4.1.3.2.4.3 Test bitstreams path preserving #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.

Files:

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

4.2.4.1.3.2.4.4 Test bitstreams path preserving #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.

Files:

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

4.2.4.1.3.2.5 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.

4.2.4.1.3.3 Position interpolator

4.2.4.1.3.3.1 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 14496-11:2005 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 14496-11:2005 shall also be implemented.

In the following Subclauses the term ‘reference decoder’ means the technical report software verifier (ISO/IEC 14496-5:2001/Amd.7:2005). The reference decoder is a decoder that implements precisely the decoding process as specified in ISO/IEC 14496-11:2005.

4.2.4.1.3.3.2 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 4.2.4.1.3.3.3 and 4.2.4.1.3.3.4.

4.2.4.1.3.3.3 Test bitstreams for key preserving

4.2.4.1.3.3.3.1 Test bitstreams key preserving #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.

Files:

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	pi_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

4.2.4.1.3.3.3.2 Test bitstreams key preserving #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.

Files:

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	bAllSameMantissaDigitFlag 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	bAllSameExponentSign 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

4.2.4.1.3.3.3.3 Test bitstreams key preserving #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.

Files:

Name	Attribute	Bitstream info. for key	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
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 2^{nd} order	pi_kp#3-2ndDPCM	pi_kp#3-2ndDPCM
pi_kp#3-7	shift operation	bShiftFlag is enabled	pi_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

4.2.4.1.3.3.3.4 Test bitstreams key preserving #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.

Files:

Name	Attribute	Bitstream info. for key value	Bitstream (.mp4)	Reference file (.dec - Interpolator only)
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

4.2.4.1.3.3.4 Test bitstreams for path preserving

4.2.4.1.3.3.4.1 Test bitstreams path preserving #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.

Files:

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

4.2.4.1.3.3.4.2 Test bitstreams path preserving #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.

Files:

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

4.2.4.1.3.3.4.3 Test bitstreams path preserving #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.

Files:

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

4.2.4.1.3.3.4.4 Test bitstreams path preserving #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.

Files:

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	nKVDPCMOrders 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	bIsUnaryAACs are disabled for all components	pi_pp#4-binaryaac	pi_pp#4-binaryaac
pi_pp#4-5	Unary AAC decoding	bIsUnaryAACs 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

4.2.4.1.3.3.5 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.

4.2.4.2 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 Bone-Based Animation (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/Amd7:2005 with source code available in electronic format.

4.2.4.2.1 Bitstream conformance

4.2.4.2.1.1 Conformance requirements

BIFS streams shall comply with the specifications in 5.4 of ISO/IEC 14496-16:2006.

4.2.4.2.1.2 Measurement procedure

Syntax of the BIFS stream shall meet the requirements of 5.4 of ISO/IEC 14496-16:2006.

4.2.4.2.1.3 Tolerance

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

4.2.4.2.2 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.

Name	Attributes	Elementary bitstream (.bba)	Bitstream (.mpa)
BBA_Bones_DCT01	Contains animation only for the bones. Compressed by DCT with quantization step 1.	BBA_Bones_DCT01	BBA_Bones_DCT01
BBA_Bones_DCT12	Contains animation only for the bones. Compressed by DCT with quantization step 12.	BBA_Bones_DCT12	BBA_Bones_DCT12
BBA_Bones_DCT31	Contains animation only for the bones. Compressed by DCT with quantization step 31.	BBA_Bones_DCT31	BBA_Bones_DCT31
BBA_Bones_PRE01	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 1.	BBA_Bones_PRE01	BBA_Bones_PRE01
BBA_Bones_PRE12	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 12.	BBA_Bones_PRE12	BBA_Bones_PRE12
BBA_Bones_PRE31	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 31.	BBA_Bones_PRE31	BBA_Bones_PRE31
BBA_BoneAndMuscle_DCT01	Contains animation for bones and muscles. Compressed by DCT with quantization step 1.	BBA_BM_DCT01	BBA_BM_DCT01
BBA_BoneAndMuscle _DCT12	Contains animation for bones and muscles. Compressed by DCT with quantization step 12.	BBA_BM_DCT12	BBA_BM_DCT12
BBA_BoneAndMuscle _DCT31	Contains animation for bones and muscles. Compressed by DCT with quantization step 31.	BBA_BM_DCT31	BBA_BM_DCT31
BBA_BoneAndMuscle_PRE01	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 1.	BBA_BM_PRE01	BBA_BM_PRE01
BBA_BoneAndMuscle_PRE12	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 12.	BBA_BM_PRE12	BBA_BM_PRE12
BBA_BoneAndMuscle_PRE31	Contains animation for bones and muscles. Compressed by predictive coder with quantization step 31.	BBA_BM_PRE31	BBA_BM_PRE31

4.2.4.3 BBA stream updates for morph-based animation

The BBA stream syntax was updated in ISO IEC 14496-16:2006 in order to represent in a BBA animation frame, the values of the weights associated to each target shape in a MorphShape node. The morph is the third surface deformer that can be represented in a BBA data (with bones and muscles). A BBA animation frame can contain data for several MorphShape nodes.

4.2.4.3.1 Conformance points

4.2.4.3.1.1 Covered functionalities

The following Subclauses specify the normative tests for verifying conformance of BBA compressed bitstream and BBA decoder. Those normative tests make use of test data (bitstream test suites).

4.2.4.3.2 Bitstream conformance

4.2.4.3.2.1 Conformance requirements

BBA streams shall comply with the specifications for BBA compression in ISO/IEC 14496-16:2006 Texture and Morphing.

4.2.4.3.2.2 Measurement procedure

Syntax of the BBA stream shall meet the requirements of BBA compression in ISO/IEC 14496-16:2006 Texture and Morphing.

4.2.4.3.2.3 Tolerance

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

4.2.4.3.3 Terminal conformance

4.2.4.3.3.1 Conformance requirements

A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-16:2006 Morphing and Texture 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 14496-11:2005 shall also be implemented.

4.2.4.3.3.2 Test bitstreams

Name	Attributes	Bitstream (.mp4)	Reference file
MorphAnim	Animation of a morph space with one base shape and four target shapes consisting in linear transitions between the targets.	cube2sphere_morph_anim	cube2sphere_morph_anim

4.2.4.3.3.3 Measurement procedure

The terminal should display an animation consisting in the deformation of a textured object.

4.2.4.3.3.4 Tolerance

There is no tolerance for deforming the object.

4.2.4.4 Frame-based animated mesh compression

4.2.4.4.1 Conformance Points

4.2.4.4.1.1 Covered functionalities

The conformance points for Frame-based Animated Mesh Compression (FAMC) cover:

- The compression of different attributes per vertex,
- Different compression configurations (DCT, Lift, Layer-based Decomposition) and possible combinations between,
- CABAC-based arithmetic encoding,
- Temporal, spatial and SNR scalability.

The following Subclauses specify the normative tests for verifying conformance of FAMC. Those normative tests make use of test data (bitstream test suites).

FAMC is using the BitWrapper node in a separate bitstream referred by a url.

The following Subclauses specify the normative tests for verifying conformance of FAMC compressed bitstreams and FAMC decoder. Those normative tests make use of test data (bitstream test suites).

4.2.4.4.2 Bitstream conformance

4.2.4.4.2.1 Conformance requirements

BIFS streams shall comply with the specifications of 4.3.6 of ISO/IEC 14496-16:2006/Amd2:2009 and 7.2.2.23 of ISO/IEC 14496-11:2005.

4.2.4.4.2.2 Measurement procedure

BIFS streams shall comply with the specifications of 4.3.6 of ISO/IEC 14496-16:2006/Amd2:2009 and 7.2.2.23 of ISO/IEC 14496-11:2005.

4.2.4.4.2.3 Tolerance

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

4.2.4.4.3 Terminal conformance

4.2.4.4.3.1 Conformance requirements

A compliant decoder shall implement a decoding process that is equivalent to the one specified in ISO/IEC 14496-16:2006/Amd2:2009 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 specified in Clause 8 of ISO/IEC 14496-11:2005 shall also be implemented.

4.2.4.4.3.2 Test bitstreams

Name	Attribute	Bitstream (.mp4)	Reference file (.wrl)
DCT_Troll	DCT-based decoding of an animation with time varying geometry and normals. Encoding parameters described in "Troll\DCT\NoScalability\troll_Encoding_log.wrl"	Troll\DCT\NoScalability\troll-url	Troll\OriginalAnimation\troll_#
DCT_Quality_Scalability_Troll	Quality scalable DCT-based decoding of an animation with time varying geometry and normals. Encoding parameters described in "Troll\DCT\QualityScalability\troll_Encoding_log.wrl"	Troll\DCT\QualityScalability\troll-url	Troll\OriginalAnimation\troll_#
LIFT_Troll	LIFT-based decoding of an animation with time varying geometry and normals. Encoding parameters described in "Troll\LIFT\NoScalability\troll_Encoding_log.wrl"	Troll\LIFT\NoScalability\troll-url	Troll\OriginalAnimation\troll_#
LIFT_Quality_Scalability_Troll	Quality scalable LIFT-based decoding of an animation with time varying geometry and normals. Encoding parameters described in "Troll\LIFT\QualityScalability\troll_Encoding_log.wrl"	Troll\LIFT\QualityScalability\troll-url	Troll\OriginalAnimation\troll_#
LD_NoScalability_Troll	Non-scalable decoding of an animation with time varying geometry and normals using LD. LD-based encoded bit stream without scalability (one spatial layer, I and P-frames). Encoding parameters described in "MPEG-4\Test\AFX\FAMC\Troll\LD\NoScalability\troll_Encoding_log.wrl"	MPEG-4\Test\AFX\FAMC\Troll\LD\NoScalability\troll-url	Troll\OriginalAnimation\troll_#
LD_SpatialScalability_Troll	Spatially scalable decoding of an animation with time varying geometry and normals using LD. LD-based encoded bit stream with spatial scalability (16 spatial layers, I and P-frames). Encoding parameters described in "Troll\LD\Spatial_Scalability\troll_Encoding_log.wrl"	Troll\LD\Spatial_Scalability\troll-url	Troll\OriginalAnimation\troll_#
LD_SpatialAndTemporalScalability_Troll	Spatially and temporally scalable decoding of an animation with time varying geometry and normals using LD. LD-based encoded bit stream with spatial and temporal scalability (16 spatial layers, I, P and (hierarchical) B-frames). Encoding parameters described in "MPEG-4\Test\AFX\FAMC\Troll\LD\Spatial&Temporal_Scalability\troll_Encoding_log.wrl"	MPEG-4\Test\AFX\FAMC\Troll\LD\Spatial&Temporal_Scalability\troll-url	Troll\OriginalAnimation\troll_#
DCT_LD_SpatialAndQualityScalability_Troll	Spatial and quality scalable DCT-LD-based decoding of an animation with time varying geometry and normals. Encoding parameters described in "MPEG-4\Test\AFX\FAMC\Troll\DCT+LD\Spatial&QualityScalability\troll_Encoding_log.wrl"	Troll\DCT+LD\Spatial&QualityScalability\troll	Troll\OriginalAnimation\troll_#
Lift_LD_SpatialAndQualityScalability_Troll	Spatial and quality scalable Lift-LD-based decoding of an animation with time varying geometry and normals. Encoding parameters described in "MPEG-4\Test\AFX\FAMC\Troll\Lift+LD\Spatial&QualityScalability\troll_Encoding_log.wrl"	Troll\Lift+LD\Spatial&QualityScalability\troll	Troll\OriginalAnimation\troll_#