INTERNATIONAL STANDARD

ISO/IEC 14496-16

Fourth edition
2011-11-01

AMENDMENT 2
2014-01-15

Information technology — Coding of audio-visual objects —

Part 16: **Animation Framework eXtension (AFX)**

AMENDMENT 2: Multi-resolution 3D mesh compression

Technologies de l'information — Codage des objets audiovisuels —

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

AMENDEMENT 2: Compression de maillages 3D multirésolution

Reference number
ISO/IEC 14496-16:2011/Amd.2:2014(E)

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

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

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

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

Amendment 2 to ISO/IEC 14496-16:2011 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 16: Animation Framework eXtension (AFX)

AMENDMENT 2: Multi-resolution 3D mesh compression

Add 5.2.6:

5.2.6 Multi-Resolution 3D Mesh Coding

Multi-Resolution 3D Mesh Coding (MR3DMC) specifies a progressive compression approach for manifold triangular 3D meshes providing efficient rate-distortion performances and supporting the following functionalities:

- Lossless connectivity coding: retrieve the original connectivity with a possible permutation of the mesh vertices/triangles.

- Spatial scalability: the mesh resolution (i.e. the number of triangles/vertices) is adapted to the terminal rendering performances and to the available bandwidth.

- Quality scalability: the precision of coordinates/attributes is progressively refined as the bitstream is decoded.

- Near-lossless encoding of the geometry/attributes: the maximal error permitted, when the entire bitstream is decoded, is controlled by varying a set of encoder parameters.

5.2.6.1 MR3DMC Bistream structure

The MR3DMC stream describes a multi-resolution representation of any triangular mesh stored as an IndexedFaceSet, with single or multiple attributes defined per vertex of per triangle. The stream is decoded in a coarse-to-fine way by exploiting a set of spatial and quality levels of detail (LODs). The bitstream is composed of two main components (*cf.* Figure AMD1.1):

- The header: describing general information about the coded mesh.

- The data stream: describing the mesh LODs.

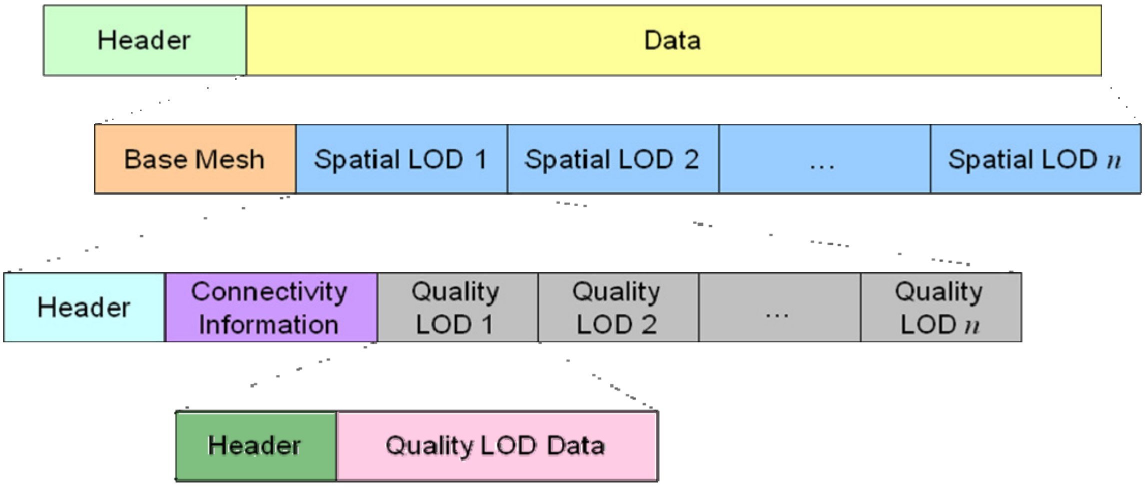

Figure AMD1.1 — MR3DMC stream structure

The MR3DMCStream is encapsulated in an AFX stream and has the following AFX object code:

Table AMD1.1 — AFX object code

AFX object code	Object	Associated node	Type value of bitwrapper
0x0C	Multi-Resolution 3D Mesh	IndexedFaceSet	2

When used in a BIFS scene, the value of the field "type" is 2.

5.2.6.2 MR3DMC Bitstream syntax and semantics

5.2.6.2.1 MR3DMC Bistream structure

5.2.6.2.1.1 Syntax

```
class MR3DMCStream {
      MR3DMCStreamHeader header;
      MR3DMCStreamData data;
}
```

5.2.6.2.1.2 Semantics

header: This is the header buffer of MR3DMC.

data: This is the data buffer of MR3DMC.

5.2.6.2.2 MR3DMCStreamHeader class

5.2.6.2.2.1 Syntax

```
class MR3DMCStreamHeader {

      unsigned int (32) streamSize;
```

```
bit (8) simplificationMode;
bit (16) numberOfSpatialLODs;
float (32) creaseAngle;
bit (1) ccw;
bit (1) solid;
bit (1) convex;
bit (1) colorPerVertex;
bit (1) normalPerVertex;
bit (1) otherAttributesPerVertex;
bit (1) isTriangularMesh;
bit (1) markerBit // always set as 1

unsigned int (32) numberOfCoord;
unsigned int (32) numberOfNormal
unsigned int (32) numberOfTexCoord;
unsigned int (32) numberOfColor;
unsigned int (32) numberOfOtherAttributes;
unsigned int (32) maxNumberOfVerticesInPatch


if (numberOfOtherAttributes >0) {
    unsigned int (8) dimensionOfOtherAttributes;
}
if (numberOfCoord>0) {
    unsigned int (32) numberOfCoordIndex;
    bit(8) QPforGeometry;
}
if (numberOfNormal>0) {
    bit(8) QPforNormal;
}
if (numberOfColor>0) {
    bit(8) QPforColor;
}
if (numberOfTexCoord>0) {
    bit(8) QPforTexCoord;
}
if (numberOfOtherAttributes >0) {
    bit(8) QPforOtherAttributes
}
if (numberOfCoord>0) {
    for(i=0;i<3;i++) {
        float(32) quantMinGeometry[i];
        float(32) quantRangeGeometry[i];
    }
}
if (numberOfNormal>0) {
    for (i=0;i<3;i++) {
        float(32) quantMinNormal[i];
        float(32) quantRangeNormal[i];
    }
}
if(numberOfColor>0) {
    for(i=0;i<3;i++) {
        float(32) quantMinColor[i];
        float(32) quantRangeColor[i];
    }
}
if(numberOfTexCoord>0) {
    for(i=0;i<2;i++) {
        float(32) quantMinTexCoord[i];
```

```
                float(32) quantRangeTexCoord[i];
            }
        }
        if(numberOfOtherAttributes>0) {
            for(i=0;i< dimensionOfOtherAttributes;i++) {
                float(32) quantMinOtherAttributes[i];
                float(32) quantRangeOtherAttributes[i];
            }
        }
        unsigned int (32) numberOfConnectedComponents;
};
```

5.2.6.2.2.2 Semantics

streamSize: A 32-bit unsigned integer describing the size in bytes of the current MR3DMC stream.

simplificationMode: A 8-bit unsigned integer indicating the simplification strategy

Table AMD1.2 — MR3DMC simplification modes

simplificationMode	Method
0	Geometry aware simplification
1	Connectivity-based simplification
2-255	ISO reserved

numberOfSpatialLODs: A 16-bit unsigned integer indicating the number of spatial LODs

creaseAngle: A 32-bit float indicating the IFS *creaseAngle* parameter which controls the default normal generation process.

ccw: 1-bit flag describing the IFS *ccw* parameter, which indicates whether the vertices are ordered in a counter-clockwise direction when the mesh is viewed from the outside.

solid: 1-bit flag describing the IFS *solid* parameter which indicates whether the shape encloses a volume.

convex: 1-bit flag describing the IFS *solid* parameter which indicates whether all faces in the shape are convex (should be always 1 for triangular meshes).

colorPerVertex: 1-bit flag describing the IFS *colorPerVertex* parameter which indicates whether the colors are defined per vertex.

normalPerVertex: 1-bit flag describing the IFS *normalPerVertex* parameter which indicates whether the normals are defined per vertex.

otherAttributesPerVertex: 1-bit flag describing whether the other attributes are defined per vertex.

isTriangularMesh: 1-bit flag describing whether the mesh is triangular (should be always 1).

markerBit: Always set as 1

numberOfCoord: A 32-bit unsigned integer indicating the number of position coordinates in the fine resolution mesh.**numberOfNormal**: A 32-bit unsigned integer indicating the number of normal coordinates in the fine resolution mesh.

numberOfTexCoord: A 32-bit unsigned integer indicating the number of texture coordinates in the fine resolution mesh.

numberOfColor: A 32-bit unsigned integer indicating the number of color coordinates in the fine resolution mesh.

numberOfOtherAttributes: A 32-bit unsigned integer indicating the number of the other attributes in the fine resolution mesh.

maxNumberOfVertiexInPatch: A 32-bit unsigned integer indicating the maximum number of vertices in the patches, which the spatial LOD consists of.

dimensionOfOtherAttributes: A 32-bit unsigned integer indicating the dimension (i.e., number of attributes) of the other attributes.

numberOfCoordIndex: A 32-bit unsigned integer indicating the number of faces associated to the position coordinates.

QPforGeometry: A 8-bit data indicating quantization parameter for geometry.

QPforNormal: A 8-bit data indicating quantization parameter for normals.

QPforColor: A 8-bit data indicating quantization parameter for colour.

QPforTexCoord: A 8-bit data indicating quantization parameter for texture coordinate.

QPforOtherAttributes: A 8-bit data indicating quantization parameter for other attributes

quantMinGeometry[]:1 by 3 array containing 32 bit floating data indicating minimum value used for geometry quantization

quantRangeGeometry[]: 1 by 3 array containing 32-bit floating point data indicating range values used for geometry quantization

quantMinNormal[]: 1 by 3 array containing 32 bit floating data indicating minimum value used for normal quantization

quantRangeNormal[]: 1 by 3 array containing 32-bit floating point data indicating range values used for normal quantization

quantMinColor[]: 1 by 3 array containing 32 bit floating data indicating minimum value used for color quantization

quantRangeColor[]: 1 by 3 array containing 32-bit floating point data indicating range values used for color quantization

quantMinTexCoord[]: 1 by 2 array containing 32 bit floating data indicating minimum value used for texcoord quantization

quantRangeTexCoord[]: 1 by 2 array containing 32-bit floating point data indicating range values used for texcoord quantization

quantRangeOtherAttributes[]: 1 by dimensionOfOtherAttributes array containing 32-bit floating point indicating range values used for normal quantization

numberOfConnectedComponents: A 32-bit unsigned integer indicating the number of the connected components.

5.2.6.2.3 MR3DMCStreamData class

5.2.6.2.3.1 Syntax

```
class MR3DMCStreamData {
    unsigned int (32) numberOfBaseVertices;
    unsigned int (32) numberOfBaseTriangles;
    TFANIndexDecoder(3, numberOfBaseVertices, numberOfBaseTriangles, 0, 0) baseMesh;
    for (int layer = 1; layer < numberOfSpatialLODs; layer++) {
        SpatialLODDecoder spatialLODdecoder;
    }
}
```

5.2.6.2.3.2 Semantics

numberOfBaseVertices: A 32-bit unsigned integer indicating the number of vertices in the base mesh.

numberOfBaseTriangles: A 32-bit unsigned integer indicating the number of triangles in the base mesh.

baseMesh: A TFAN strearm (cf. 5.2.5.3.9) describing the base mesh.

spatialLODdecoder: A stream specifying a spatial LOD layer.

5.2.6.2.4 SpatialLODDecoder Class

5.2.6.2.4.1 Syntax

```
class SpatialLODDecoder {
    SpatialLODDecoderHeader spatialLODDecoderHeader;
    ConnectivityLODDecoder connectivityLODDecoder;
    for (int qlayer = 0; qlayer < numberOfQualityLODs; qlayer++) {
        QualityLODDecoder qualityLODDecoder;
    }
}
```

5.2.6.2.4.2 Semantics

spatialLODDecoderHeader: Header buffer specifying the current spatial LOD properties.

connectivityLODDecoder: A ConnectivityLODDecoder stream describing the connectivity information of the current LOD.

qualityLODDecoder: A QualityLODDecoder stream describing the geometry refinement information for the different LODs associated with the current spatial LOD.

5.2.6.2.5 SpatialLODHeaderDecoder Class

5.2.6.2.5.1 Syntax

```
class SpatialLODHeaderDecoder
{
    bit (32) numberOfVertices;
    bit (32) numberOfTriangles;
    bit (8) numberOfQualityLayers;
    bit (8) qualityLODEncodingStrategy
}
```

5.2.6.2.5.2 Semantics

numberOfVertices: A 32 bit integer specifying the number of vertices in the current spatial LOD.

numberOfTriangles: A 32 bit integer specifying the number of triangles in the current spatial LOD.

numberOfQualityLODs: A 32 bit integer specifying the number of quality LODs in the current spatial LOD.

qualityLODEncodingStrategy: A 8-bit unsigned integer indicating the quality LOD encoding strategy

5.2.6.2.6 ConnectivityLODDecoder Class

5.2.6.2.6.1 Syntax

```
class ConnectivityLODDecoder
{
    if ( simplificationMode == 0) {
        TFANIndexDecoder(3, numberOfVertices, numberOfTriangles, 0, 0) connectivityCurrentLOD;
        IntArrayDecoder(numberOfVertices,1)    decodedVertexMapping;
    }
    else {
        for ( cc=0; cc < numberOfConnectedComponents; ++cc ) {
            unsigned int (32) numberOfRefinedVertices[cc]    = 0;
        }
        for ( cc = 0; cc < numberOfConnnectedComponents; ++cc ) {
            unsigned int(32) numberOfConnectivitySymbols;
            bit(8) conquestMode
            for ( int v = 0; v < numberOfConnectivitySymbols; ++v) {
                valenceOfRefinedVertices[v] = arithmetic_decoder.decode();
                if ( valenceOfRefinedVertices[v] > 0 ) {
                    numberOfRefinedVertices[cc]++;
                }
            }
        }
    }
}
```

5.2.6.2.6.2 Semantics

connectivityCurrentLOD: A TFANIndexDecoder stream (cf. 5.2.5.3.9s) describing the connectivity of the current spatial LOD.

decodedVertexMapping: A bitstream of type IntArrayDecoder specifying for each vertex of the previous LOD its index the current one.

numberOfConnectivitySymbols: An array of 32 bit unsigned integer specifying the number of connectivity symbols, which represent the valences of refined vertices or null in the current spatial LOD for each connected component.

conquestMode: A 8-bit unsigned integer indicating conquest strategy in the current spatial LOD.

Table AMD1.3 — MR3DMC conquest modes

conquestMode	Method
0	Decimating Conquest
1	Cleaning Conquest
2-255	ISO reserved

valenceOfRefinedVertices: An array of integer describing the valences of inserted vertices or null symbols in order to refine the previous spatial LOD.

numberOfRefinedVertices: An array of 32-bit unsigned integer indicating the number of the refined vertices for each spatial LOD

5.2.6.2.7 QualityLODDecoder class

5.2.6.2.7.1 Syntax

```
class QualityLODDecoder {
    QualityLODHeader qualityLODMR3DMCHeader
    QualityLODData qualityLODMR3DMCData

}
```

5.2.6.2.7.2 Semantics

qualityLODHeader: Header buffer specifying the current quality LOD properties.

qualityLODData: A QualityLODData stream describing the compressed predicted approximation errors associated with the current quality LOD vertices (cf. Annex U.7)

5.2.6.2.8 QualityLODHeader class

5.2.6.2.8.1 Syntax

```
class QualityLODHeader {
    if (qualityLODEncodingStrategy == 0 ) {
        bits(8) numberOfCurrentQPLayer;
        bits(32) numberOfVerticesOfCurrentQualityLOD;
    }
}
```

5.2.6.2.8.2 Semantics

numberOfCurrentQPLayer: An 8 bit integer indicating the number of QP layer in the current quality LOD.

numberOfVerticesOfCurrentQualityLOD: An 32 bit integer indicating the number of vertices in current quality LOD.