INTERNATIONAL STANDARD ISO/IEC 14496-16:2004
TECHNICAL CORRIGENDUM 1

Published 2005-03-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Coding of audio-visual objects —

Part 16:
Animation Framework eXtension (AFX)

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Codage des objets audiovisuels —

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

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 14496-16:2004 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/Cor1:2005
Withdrawn

ICS 35.040 **Ref. No. ISO/IEC 14496-16:2004/Cor.1:2005(E)**

© ISO/IEC 2005 – All rights reserved

Published in Switzerland

In Clause 3, replace the list with:

AFX Animation Framework eXtension

BIFS BInary Format for Scene

DIBR Depth-Image Based Representation

ES Elementary Stream

IBR Image-Based Rendering

NDT Node Data Type

OD Object Descriptor

VRML Virtual Reality Modelling Language

In subclause 4.3.1, remove reference to particle system:

- Particle systems, which consists of the following nodes: **Particles**, **ParticleInitiBox**, **ParticleInitCone**, **ParticleObstacle**, and **PointAttractor**.

In subclause 4.3.4.1.3, replace:

A MeshGrid mesh allows view-dependent streaming (see subclause 5.1) and particular animation possibilities, such as: hierarchical grid animation, and offset based animation (see subclause 4.3.4.3).

with:

A MeshGrid mesh allows view-dependent streaming (see subclause 5.2) and particular animation possibilities, such as: hierarchical grid animation, and offset based animation (see subclause 4.3.4.3).

In subclause 4.4.2.1, replace Table 2: General arithmetic operator with:

Op	Description	Rules for each P(X1,X2,X3,X4)	Syntax
Sadd	Arithmetic addition of the density of two forms	$d_r(P) = d_0(P) + d_1(P)$	**Sadd (F0, F1)**
Smul	Arithmetic multiplication of the density of two forms	$d_r(P) = d_0(P) * d_1(P)$	**Smul (F0, F1)**
Sdif	The positive difference of two forms *F0* and *F1*.	Returns the difference between the densities if this result is nonnegative, and 0 if the result is negative.	**Sdif (F0, F1)**
Sexp	Exponentiation of forms	Raises one form to the power of another form. The density of a point with respect to *F1* serves as exponent to the density of the same point with respect to *F0*.	**Sexp (F0, F1)**
Sgcd	Greatest common divisor	Returns the *gcd* of the densities of two forms.	**Sgcd (F0, F1)**
Slcm	Least common multiple	Returns the *lcm* of the densities of two forms.	**Slcm (F0, F1)**
Smod	Integral remainder	The Integral remainder operator calculates the integer remainder when the density of a point with respect to *F0* is divided by the density of that point with respect to *F1*.	**Smod (F0, F1)**
Ssab	Absolute difference	Returns the result of the subtraction between the densities of two forms when this result is nonnegative. Otherwise, the return is the result opposed value.	**Ssab (F0, F1)**

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

Scub	Integral cube root	Returns the integral cube root of the density of the form.	**Scub (F0)**
Ssqr	Integral square root	Returns the integral square root of the density of the form.	**Ssqr (F0)**
Smax	Maximum	This operator is equivalent to the ternary union for n-ary logic.	**Smax (F0, F1)**
Smin	Minimum	This operator is equivalent to the ternary intersection for n-ary logic.	**Smin (F0, F1)**

In subclause 4.4.2.2, replace Table 3: Ternary logic operators with:

Op	Description	Rules for each P(X1,X2,X3,X4)	Syntax
Suni	Ternary union of two forms.	**F1** **Suni** | 0 | 1 | 2 0 | 0 | 1 | 2 **F0** 1 | 1 | 1 | 2 2 | 2 | 2 | 2	**Suni (F0, F1)**
Sint	Ternary intersection	**F1** **Sint** | 0 | 1 | 2 0 | 0 | 0 | 0 **F0** 1 | 0 | 1 | 1 2 | 0 | 1 | 2	**Sint (F0, F1)**
Simp	Ternary implication	**F1** **Simp** | 0 | 1 | 2 0 | 2 | 2 | 2 **F0** 1 | 1 | 1 | 2 2 | 0 | 1 | 2	**Simp (F0, F1)**
Simr	Reciprocal Ternary implication	**F1** **Simr** | 0 | 1 | 2 0 | 2 | 1 | 0 **F0** 1 | 1 | 1 | 1 2 | 0 | 1 | 2	**Simr (F0, F1)**
Sdua	Ternary dual of the volume	**Sdua** | 0 | 2 **F0** 1 | 1 2 | 0	**Sdua (F0)**

In subclause 4.4.2.3, replace Table 4: Filtering and test operators with:

Filter	Description	Rules	Syntax
Seqf	Equality filter	The density of the volume that checks the equality test, and 0 otherwise.	**Seqf (F0, F1)**
Sgtef	greater than or equal filter	The density of the second volume if the density of the first one is greater than or equal to the density of the second volume, and 0 otherwise.	**Sgtef (F0, F1)**
Sgtf	Greater than filter	The density of the second volume if the density of the first one is greater than the density of the second volume, and 0 otherwise.	**Sgtf (F0, F1)**
Sltef	Less than or equal filter	The density of the second volume if the density of the first one is less than or equal to the density of the second volume, and 0 otherwise.	**Sltef (F0, F1)**
Sltf	Less than filter	The density of the second volume if the density of the first one is less than the density of the second volume, and 0 otherwise.	**Sltf (F0, F1)**

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

Sevnf	Even filter	The volume density if the density is even, and 0 otherwise.	**Sevnf (F0)**
Soddf	Odd filter	The result is the volume density if the density is odd, and 0 otherwise.	**Soddf (F0)**
Sneqf	Difference filter	The result is the second volume density if the densities are different, and 0 otherwise.	**Sneqf (F0, F1)**

In subclause 4.4.3, replace old operators symbols by the new ones. Replace:

```
# b: egg cut by a box
Group{
    children [
        # Primary SolidRep
        DEF Le_Solid3 SolidRep{
            bboxSize 5 5 5
            densityList [1 3 4] # choice of densities to display
            dual FALSE
        }
        DEF Le_Script Script{ # Solid Tree definition
            field SFNode Srepout USE Le_Solid3
            #primitives
            field SFNode White # container
                Transform {
                translation 0 0 0
                    children [
                        Shape {
                        …
                        geometry
                            Quadric { P0 1 0 0 1
                            P1 -1 0 0 1
                            P2 0 1 0 0
                            P3 0 0 1 0
                            P4 0 1 0 1
                            P5 0 0 1 1
                            boxSize 1 1.3 0.95
                            solid FALSE}
                        }
                    ]
                }
            field SFNode Yolk # matter inside
                Transform {
                    children [
                        Shape {
                        …
                        geometry
                            Quadric { P0 1 0 0 1
                            P1 -1 0 0 1
                            P2 0 1 0 0
                            P3 0 0 1 0
                            P4 0 1 0 1
                            P5 0 0 1 1
                            bboxSize 0.7 0.7 0.4
                            density 3
                            solid FALSE}
                        }
                    ]
                }
            field SFNode CuttingBox # cutting tool
                Transform {
                translation 0.5 0 0
                    children [
                        Quadric { P0 -1 0 0 1
                        P1 1 0 0 1
                        P2 0 1 0 0
                        P3 0 0 1 0
                        P4 0 1 0 0
                        P5 0 0 1 0
                        bboxSize 1 1 1
                        density 4
                        solid FALSE}
                    ]
                }
            directOutput TRUE
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

```
            url "vrmlscript:
                function initialize(){
                    Srepout.solidTree = (White + Yolk)- CuttingBox;}
            "
        }
    ]
}
```

with:

```
# b: egg cut by a box
Group{
    children [
        # Primary SolidRep
        DEF Le_Solid3 SolidRep{
            bboxSize 5 5 5
            densityList [1 3 4] # choice of densities to display
            dual FALSE
        }
        DEF Le_Script Script{ # Solid Tree definition
            field SFNode Srepout USE Le_Solid3
            #primitives
            field SFNode White # container
                Transform {
                translation 0 0 0
                    children [
                        Shape {
                        …
                        geometry
                            Quadric { P0 1 0 0 1
                            P1 -1 0 0 1
                            P2 0 1 0 0
                            P3 0 0 1 0
                            P4 0 1 0 1
                            P5 0 0 1 1
                            boxSize 1 1.3 0.95
                            solid FALSE}
                        }
                    ]
                }
            field SFNode Yolk # matter inside
                Transform {
                    children [
                        Shape {
                        …
                        geometry
                            Quadric { P0 1 0 0 1
                            P1 -1 0 0 1
                            P2 0 1 0 0
                            P3 0 0 1 0
                            P4 0 1 0 1
                            P5 0 0 1 1
                            bboxSize 0.7 0.7 0.4
                            density 3
                            solid FALSE}
                        }
                    ]
                }
            field SFNode CuttingBox # cutting tool
                Transform {
                translation 0.5 0 0
                    children [
                        Quadric { P0 -1 0 0 1
                        P1 1 0 0 1
                        P2 0 1 0 0
                        P3 0 0 1 0
                        P4 0 1 0 0
                        P5 0 0 1 0
                        bboxSize 1 1 1
                        density 4
                        solid FALSE}
                    ]
                }
            directOutput TRUE
            url "vrmlscript:
                function initialize(){
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

```
                Srepout.solidTree = Sdif(Sadd(White,Yolk), CuttingBox);}
            "
        }
    ]
}
```

In subclause 4.5.1, replace:

Two technologies are defined in this subclause:
1. Depth image-based representation comprises the following nodes: **DepthImage**, **SimpleTexture**, **PointTexture**, and **OctreeImage**.
2. Light-field mapping comprises the following nodes: **LFM_Appearance**, **LFM_FrameList**, **LFM_LightMap**, **LFM_SurfaceMapList**, **LFM_ViewMapList**, and **LFM_BlendList**.

with:

The technologie defined in this subclause is called "Depth image-based representation" and comprises the following nodes: **DepthImage**, **SimpleTexture**, **PointTexture**, and **OctreeImage**.

In subclause 4.5.3, Light Field Mapping, remove the entire subclause.

In subclause 4.7.1.1.1, Node Interface, replace Types of the SBBone:

SBBone{ #%NDT=SFSBBoneNode

with:

SBBone{ #%NDT=SFSBBoneNode, SF3DNode, SF2DNode

In subclause 4.7.1.1.1, Node Interface, replace default value for the "scale" field:

exposedField SFVec3f scale 0 0 0

with:

exposedField SFVec3f scale 1 1 1

In subclause 4.7.1.2.1, Node Interface, replace Types of the SBSegment:

SBSegment{ #%NDT=SFSBSegmentNode

with:

SBSegment{ #%NDT=SFSBSegmentNode, SF3DNode, SF2DNode

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

In subclause 4.7.1.3.1, Node Interface, replace Types of the SBSite:

SBSite { #%NDT=SFSBSiteNode

with:

SBSite { #%NDT=SFSBSiteNode, SF3DNode, SF2DNode

In subclause 4.7.1.4.1, Node Interface, replace Types of the SBMuscle:

SBMuscle{ #%NDT=SFSBMuscleNode

with:

SBMuscle{ #%NDT=SFSBMuscleNode, SF3DNode, SF2DNode

In subclause 5.4.2.8.1, replace the definition of the bba_object_plane_mask class:

```
class bba_object_plane_mask() {
        bit(5) NumberOfInterpolatedFrames; //NIF
        if (isIntra){
                bit(5) bba_quant;
                bit(3) pow2quant;
                bit(10) NumberOfBones; //NSBB
                bit(10) NumberOfMuscles;//NSBM
                for (bone=1;bone<NumberOfBones;bone++){
                        bit(10) BoneIdentifier; //IDB
                bone_mask bnmask();
        }
        for (ms=1;ms< NumberOfMuscles;ms+){
                bit(10) MuscleIdentifier; //IDM
                bit(6) NumberControlPoints; //NCP
                bit(6) NumberKnots; //NK
                muscle_mask msmask();
        }
}
```

with:

```
class bba_object_plane_mask() {
        bit(5) NumberOfInterpolatedFrames; //NIF
        if (isIntra){
                bit(5) bba_quant;
                bit(3) pow2quant;
                bit(3) NumberOfControllerTypes;
                bit(10) NumberOfBones; //NSBB
                bit(10) NumberOfMuscles;//NSBM
                for (bone=1;bone<NumberOfBones;bone++){
                        bit(10) BoneIdentifier; //IDB
                        bone_mask bnmask();
                }
                for (ms=1;ms< NumberOfMuscles;ms+){
                        bit(10) MuscleIdentifier; //IDM
                        bit(6) NumberControlPoints; //NCP
                        bit(6) NumberKnots; //NK
                        muscle_mask msmask();
                }
}
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

In subclause 5.4.2.8.2, add at the end of the subclause the following sentence:

NumberOfControllerTypes (NMF) - a 3-bit unsigned integer indicating the number of controller types used by the current version of the BBA stream. As this version of the BBA stream use bones and muscles this variable should take value « 2 » (two).

In subclause 5.4.2.9.1, replace:

```
class bone_mask() {
  bit(1) IsTranslation_changed;
  bit(1) marker_bit;
  if (IsTranslation_changed){
    bit(1) IsTranslationOnX_changed;
    bit(1) IsTranslationOnY_changed;
    bit(1) IsTranslationOnZ_changed;
  }
  bit(1) IsRotation_changed;
  bit(1) marker_bit;
  if (IsRotation_changed){
    bit(1) IsRotationOnAxis1_changed;
    bit(1) IsRotationOnAxis2_changed;
    bit(1) IsRotationOnAxis3_changed;
  }
  bit(1) IsScale_changed;
  bit(1) marker_bit;
  if (IsScale_changed){
    bit(1) IsScaleOnX_changed;
    bit(1) IsScaleOnY_changed;
    bit(1) IsScaleOnZ_changed;
  }
  bit(1) IsScaleOrientation_changed;
  bit(1) marker_bit;
  if (IsScaleOrientation_changed){
    bit(1) IsScaleOrientation AxisX_changed;
    bit(1) IsScaleOrientation AxisY_changed;
    bit(1) IsScaleOrientation AxisZ_changed;
    bit(1) IsScaleOrientation Value_changed;
  }
  bit(1) IsCenter_changed;
  bit(1) marker_bit;
  if (IsCenter_changed){
    bit(1) IsCenterOnX_changed;
    bit(1) IsCenterOnY_changed;
    bit(1) IsCenterOnZ_changed;
  }
}
```

with:

```
class bone_mask() {
  bit(1) IsTranslation_changed;
  bit(1) marker_bit;
  if (IsTranslation_changed){
    bit(1) IsTranslationOnX_changed;
    bit(1) IsTranslationOnY_changed;
    bit(1) IsTranslationOnZ_changed;
  }
  bit(1) IsRotation_changed;
  bit(1) marker_bit;
   if (IsRotation_changed){
     bit(1) isQuaternion ;
     if (!isQuaternion){
      bit(1) IsRotationOnAxis1_changed;
      bit(1) IsRotationOnAxis2_changed;
      bit(1) IsRotationOnAxis3_changed;
```

Withdrawn
IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005

© ISO/IEC 2005 – All rights reserved

```
      }else{
    bit(1) IsRotationOnQx_changed;
    bit(1) IsRotationOnQy_changed;
    bit(1) IsRotationOnQz_changed;
    bit(1) IsRotationOnQw_changed;
     }
 }
 bit(1) IsScale_changed;
 bit(1) marker_bit;
 if (IsScale_changed){
   bit(1) IsScaleOnX_changed;
   bit(1) IsScaleOnY_changed;
   bit(1) IsScaleOnZ_changed;
 }
 bit(1) IsScaleOrientation_changed;
 bit(1) marker_bit;
 if (IsScaleOrientation_changed){
   bit(1) IsScaleOrientation AxisX_changed;
   bit(1) IsScaleOrientation AxisY_changed;
   bit(1) IsScaleOrientation AxisZ_changed;
   bit(1) IsScaleOrientation Value_changed;
 }
 bit(1) IsCenter_changed;
 bit(1) marker_bit;
 if (IsCenter_changed){
   bit(1) IsCenterOnX_changed;
   bit(1) IsCenterOnY_changed;
   bit(1) IsCenterOnZ_changed;
 }
}
```

In subclause 5.4.2.12.2, replace:

NUM_SBCs=NSBB*16+NSBM*(NCP*3+NCP+NK)

with:

$$NUM\_SBCs = \sum_{bn=1}^{NSBB} c_{bn} + \sum_{ms=1}^{NSBM} (3 * NCP_{ms} + NCP_{ms} + NK_{ms}),$$

with

$$c_{bn} = \begin{cases} 16, if & isQuaternion = 0 \\ 17, if & isQuaternion = 1 \end{cases},$$

is the number of the components for a bone transform.

In subclauses 5.1.1.2.1, Syntax of the Wavelet_Mesh_Object, replace:

```
class Wavelet_Mesh_Object {
    bit(1) WMOL;
    bit(1) isInLocalCoordinates;
    if (WMOL && WMDecoderConfig.hasScaleCoeff)
        Wavelet_Mesh_Object_Scale_Coeff Coefficients;
    ReadZT ZeroTree;
}
```

with:

```
class Wavelet_Mesh_Object {
    int(1) isInBand;
    if (isInBand)
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

```
        WMDecoderConfig WMDecoderConfig;
    bit(1) WMOL;
    bit(1) isInLocalCoordinates;
    if (WMOL && WMDecoderConfig.hasScaleCoeff)
        Wavelet_Mesh_Object_Scale_Coeff Coefficients;
    ReadZT ZeroTree;
}
```

In subclause 5.2.1, replace:

The MeshGrid stream will, in general, consist of three parts: (1) a *connectivity-wireframe description*, (2) a *reference-grid description*, and (3) a *vertices-refinement description* (i.e. refining the position of the vertices relative to the reference-grid – the offsets). A minimal stream may, however, only consist of the description of the connectivity-wireframe, which is mandatory for every stream. All of these three parts can be encoded at each resolution level, either as one single *region of interest* (ROI) or in separate ROIs, if view-dependent decoding is needed.

with:

The MeshGrid stream has a modular structure consisting of a sequence of parts. There are several types of parts with different semantics, each one identified by a unique tag. All part types are optional, and several parts of the same type can be present in the stream, but their order is not imposed. The following parts of the MeshGrid stream are encoded at each resolution level, either as one single *region of interest* (ROI) or in separate ROIs, if view-dependent decoding is needed: (1) a *connectivity-wireframe description*, (2) a *reference-grid description*, and (3) a *vertices-refinement description* (i.e. refining the position of the vertices relative to the reference-grid – the offsets).

In subclause 5.2.2.1, Global Constants, remove:

```
const unsigned char MG_SIG = 0x10;              The signature of the MeshGrid stream
```

In subclause 5.2.2.1, Global Constants, replace:

```
READ_REPOSITION_BITS                            No default value specified for the reposition bits
```

with:

```
const unsigned int READ_REPOSITION_BITS    No default value specified for the reposition bits
= 2
```

In subclause 5.2.2.2.1, Syntax for the MeshGridDecoderConfig, replace the entire section with:

```
class MeshGridDecoderConfig extends AFXExtDescriptor: bit(8) tag=1{
    // the max number of resolution levels
    unsigned int totalNumLevelsMesh;

  // number of resolution levels specified for each u,v,w direction
  // of the reference-grid
 ParsableUVW nLevelsMesh(LEVEL_BITS);
 totalNumLevelsMesh = max(nLevelsMesh.u, max(nLevelsMesh.v, nLevelsMesh.w));

  // number of slices (reference surfaces) corresponding to the last
  // resolution level, in the u,v,w directions;
 ParsableUVW nSlices(SLICES_BITS);

 // flags
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/Cor1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

```
  bit (1) hasConnectivityInfo;
  bit (1) hasRefineInfo;
  bit (1) hasRepositionInfo;
  bit (1) hasGridInfo;

  // reserved
  bit(9) attributes = 0;

  // type of homogeneous mesh
  unsigned int(2) meshType;

   if (meshType == QUADRI_MESH) {
     if (hasConnectivityInfo) {
        // connectivity bits and uniform quads splitting flags
        bit (1) sameBorderOrientation;
        bit (1) uniformSplit;
     }
     else {
        unsigned int sameBorderOrientation = 1;
        unsigned int uniformSplit = 1;
     }

     if (nSlices.u == 1 || nSlices.v == 1 || nSlices.w == 1)
      GetValue offsetAmplitude;
  }

   // the choices for cyclic mesh
   bit (3) cyclicMode;

  // number of refine bits vertex
  RefineVertexDescriptor refine;

  // filter type for grid coding
  unsigned int(FILTER_BITS) filterType;

  // scale values for the x,y,z encoded grid coordinates
  // minimum scale factor is MIN_SCALE
  ScaleXYZ gridScale(FIELD_BITS);

  // the grid corners
  GridCorners gridCorner;
}
```

In subclause 5.2.2.2.2, Semantics for the MeshGridDecoderConfig, replace the entire section with:

The **MeshGridDecoderConfig** class initializes the MeshGrid decoder. It **(1)** parses the resolution levels of the mesh (*nLevelsMesh*) for the {*u*,*v*,*w*} directions, encoded on LEVEL_BITS, with acceptable values for *nLevelsMesh.u, nLevelsMesh.v* and *nLevelsMesh.w,* lying in the range [1, 2^{31} -1], **(2)** computes the maximum number of resolution levels *totalNumLevelsMesh*, **(3)** parses the number of slices (*nSlices*) {S_U,S_V,S_W} corresponding to the last resolution level of the reference-grid, specified for the {*u*,*v*,*w*} directions, with acceptable values for *nSlices.u*, *nSlices.v* and *nSlices.w*, lying in the range [1, 2^{31} -1]**.** It reads 5 values: **(4)** 1-bit flag *hasConnectivityInfo***, (5)** 1-bit flag *hasRefineInfo*, **(6)** 1-bit flag *hasRepositionInfo*, **(7)** 1-bit flag *hasGridInfo, and* **(8)** 9-bit flag *attributes*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

Table 6 — Meaning of the flags

Flag	Meaning
hasConnectivityInfo	Boolean flag: 1. When set to '1' it indicates that the parts identified by the *MGMeshInfoTag*, *MGMeshConnectivityROInfoTag* and *MGMeshConnectivityInfoTag* tags can be present in the stream at any resolution level description. 2. When set to '0' it implies that the parts identified by the *MGMeshConnectivityROInfoTag* and *MGMeshConnectivityInfoTag* tags are not present in the stream. If the value of the *hasRefineInfo* flag is '1', then the part identified by the *MGMeshInfoTag* tag can be present in the stream only for the first resolution level description. If the *meshType* parameter defined in Table 7 has the value '2' then a default quadrilateral mesh is generated as explained in subclause 5.2.3.4.
hasRefineInfo	Boolean flag: 1. When set to '1' it indicates that the parts identified by the *MGMeshInfoTag*, *MGVerticesRefinementROIInfoTag* and *MGVerticesRefinementInfoTag* tags can be present in the stream at any resolution level description. If the value of the *hasConnectivityInfo* flag is '0' these parts can only be available for the first resolution level description, as explained in subclause 5.2.3.4. 2. When set to '0' it implies that the parts identified by the *MGVerticesRefinementROIInfoTag* and *MGVerticesRefinementInfoTag* tags are not present in the stream. The part identified by the *MGMeshInfoTag* tag can be present in the stream at any resolution level description only if the value of the *hasConnectivityInfo* flag is set to '1'.
hasRepositionInfo	Boolean flag: 1. It can be set to '1' only if the *hasConnectivityInfo* flag is also '1'. When equal to '1' it indicates that the parts identified by the *MGMeshInfoTag*, *MGVerticesRepositionROIInfoTag* and *MGVerticesRepositionInfoTag* tags can be present in the stream at any resolution level description except the last resolution level. 2. When set to '0' it implies that the parts identified by the *MGVerticesRepositionROIInfoTag* and *MGVerticesRepositionInfoTag* tags are not present in the stream. The part identified by the *MGMeshInfoTag* tag can be present in the stream at any resolution level description only if the value of one of the following two flags is set to '1': *hasConnectivityInfo*, *hasRefineInfo*.
hasGridInfo	Boolean flag: 1. When set to '1' it indicates that the parts identified by the *MGGridInfoTag*, *MGGridCoefficientsROIInfoTag* and *MGGridCoefficientsInfoTag* tags can be present in the stream at any resolution level description. 2. When set to '0' it implies that the parts identified by the *MGGridInfoTag*, *MGGridCoefficientsROIInfoTag* and *MGGridCoefficientsInfoTag* tags are not present in the stream. In this case the reference-grid points are uniformly distributed and their coordinates are computed as a linear interpolation between the eight grid corners parsed by *GridCorners* in the *MeshGridDecoderConfig* class.
attributes	A 9-bit flag reserved for future use.
uniformSplit	Boolean flag: 1. When set to '1' it indicates that the stream contains a quadrilateral mesh allowing to obtain the connectivity-wireframe for the higher resolution levels by uniformly splitting each quad recursively into four sub-quads, as illustrated in Figure 40. The parts identified by the *MGMeshInfoTag*, *MGMeshConnectivityROInfoTag* and *MGMeshConnectivityInfoTag* tags are present in the stream only at the first resolution level description. 2. When set to '0' the parts identified by the *MGMeshInfoTag*, *MGMeshConnectivityROInfoTag* and *MGMeshConnectivityInfoTag* tags can be present in the stream at any resolution level description.

Further the **MeshGridDecoderConfig** class parses **(9)** the type of the mesh (*meshType*) as explained in Table 7.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

Table 7 — Encoding of the mesh type (*meshType*)

Value	Meaning
0	GENERIC_MESH: Non-homogeneous mesh that may consist of polygons ranging from triangles to heptagons.
1	TRI_MESH: Homogeneous triangular mesh
2	QUADRI_MESH: Homogeneous quadrilateral mesh
3	HEXA_MESH: Homogeneous hexagonal mesh

If *meshType* is quadrilateral (*QUADRI_MESH*) than **(10)** a 1-bit flag (*sameBorderOrientation*) defines the number of bits used to store the connectivity links between the vertices, as explained in subclause 5.2.3.1.1, and defined in Table 8, and **(11)** another 1-bit flag (*uniformSplit*) indicates the type of multiresolution as explained in subclause 5.2.3.4 and Table 6. If the number of slices (*nSlices*) in any of the {*u,v,w*} directions is equal to '1', i.e. the reference-grid consists of one layer of points, then **(12)** the variable length variable *offsetAmplitude* defines the maximum value of the offsets as explained in subclause 4.3.4.3.2.

Table 8 — Encoding of the *sameBorderOrientation* flag

Value	Meaning
0	2-bits are used to encode a connectivity link between two vertices; it is the general case.
1	1-bit is used to encode a connectivity link between two vertices; it is a particular case which may occur for homogeneous quadrilateral meshes.

Further the **MeshGridDecoderConfig** class parses **(13)** the *cyclicMode* variable defined in Table COR1.1 that specifies the cyclic behaviour of the mesh, as explained in subclause 5.2.3.1.3, **(14)** the description of the vertices' refinement bits (*refine*), **(15)** the type of filter used in the wavelet transform (*filterType*), encoded on FILTER_TYPE bits, **(16)** the scaling factor for the {x,y,z} grid coefficients (*gridScale*), encoded on FIELD_BITS bits, **(17)** the {x,y,z} coordinates of the 8 corners of the reference-grid (*gridCorner*).

Table COR1.1 — Encoding of the *cyclicMode* variable

Value	Meaning
0	general case non-cyclic mesh.
1	cyclic mesh in the "U" direction.
2	cyclic mesh in the "V" direction.
3	cyclic mesh in the "UV" directions.
4	cyclic mesh in the "W" direction.
5	cyclic mesh in the "UW" directions.
6	cyclic mesh in the "VW" directions.
7	reserved

In subclause 5.2.2.4.2, Semantics for the MeshGridCommand, replace the entire section with:

This is an abstract base class for the different types of command units of the MeshGrid stream. This class is extended by the classes identified by the class tags defined in Table 9.

Table 9 — MeshGrid command table

Tag value	**Tag name**	**Description**
0x00	Forbidden	
0x01	MGInfoTag	Tag for the MeshGrid stream coding information.
0x02	MGMeshInfoTag	Tag for the mesh coding information for a specified mesh resolution level.
0x03	MGGridInfoTag	Tag for grid coding information for a specified grid resolution level.
0x04	MGMeshConnectivityROIInfoTag	Tag for mesh connectivity information for a specified mesh resolution level and regions of interest (ROIs) list.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/Cor 1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

0x05	MGMeshConnectivityInfoTag	Tag for mesh connectivity information for a specified mesh resolution level.
0x06	MGVerticesRepositionROIInfoTag	Tag for vertices' reposition bits (single bit-plane) for a specified mesh resolution level and regions of interest (ROIs) list.
0x07	MGVerticesRepositionInfoTag	Tag for vertices' reposition bits (single bit-plane) for a specified mesh resolution level.
0x08	MGVerticesRefinementROIInfoTag	Tag for refinement bit-planes (the offset) for a specified mesh resolution level and regions of interest (ROIs) list.
0x09	MGVerticesRefinementInfoTag	Tag for refinement bit-planes (the offset) for a specified mesh resolution level.
0x10	MGGridCoefficientsROIInfoTag	Tag for wavelet coefficients for a specified grid resolution level and tiles list.
0x11	MGGridCoefficientsInfoTag	Tag for wavelet coefficients for a specified grid resolution level.
0x12-0xFE	Reserved for ISO use	
0xFF	Forbidden	

In subclause 5.2.2.6.1, Syntax for the MGMeshDescriptor , replace the entire section with:

```
class MGMeshDescriptor(MeshGridDecoderConfig mgd) extends MGLevelDescriptor: bit(8)
tag=MGMeshInfoTag
{
    bool bLastLevel = resolutionLevel.value == mgd.totalNumLevelsMesh – 1;
    bool bReposition = mgd.hasRepositionInfo && !bLastLevel;
    bool bRefine = mgd.hasRefineInfo && (mgd.refine.bFull || bLastLevel);
    MeshDescriptor mdl[[resolutionLevel.value]](mgd.hasConnectivityInfo, bReposition, bRefine);
}
```

In subclause 5.2.2.16.1, Syntax for the MeshDescriptor, replace the entire section with:

```
// Mesh Description for level
aligned(8) class MeshDescriptor(bool hasConnectivity, bool hasReposition, bool hasRefine) {
    // some member variables
    unsigned int nBitsIndex, totalNumROIs;

    // retrieve the number of ROIs
    ParsableUVW nROIs(ROI_BITS);
    totalNumROIs = nROIs.u * nROIs.v * nROIs.w;

    // if the total number of ROIs is 0, than no mesh information
    // is available for the current resolution level
    if (totalNumROIs == 0) return;

   if (hasReposition) {
     // retrieve the reposition bits flag
     bit (2) repositionBits;
   }

    if (hasConnectivity) {
        // number of bits for the ROI indices fields
        nBitsIndex = (int) floor(log2(totalNumROIs) + 1);
```

Withdrawn
IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/Cor1:2005

© ISO/IEC 2005 – All rights reserved

```
        // number of bits for the counter fields
        unsigned int(VERTS_BITS) nBitsConnectivity;
    }

    if (hasReposition && repositionBits == READ_REPOSITION_BITS) {
        unsigned int(VERTS_BITS) nBitsReposition;
    }

    if (hasRefine) {
        unsigned int(VERTS_BITS) nBitsRefine;
    }
}
```

In subclause 5.2.2.16.2, Semantics for the MeshDescriptor, replace the entire section with:

The **MeshDescriptor** class parses the coding information for a specified mesh resolution level. It **(1)** retrieves the number of ROIs (*nROIs*) for each {*u*,*v*,*w*} direction, **(2)** computes the total number of ROIs (*totalNumROIs*). If *totalNumROIs* is equal to '0' than no mesh connectivity, reposition and refinement data is available for the current resolution level. Further, if the *hasReposition* flag is set to '1' (the *hasRepositionInfo* flag from the *MeshGridDecoderConfig* is set to '1' and the specified mesh resolution level is not the last) it reads **(3)** a 2-bit flag (*repositionBits*), which indicates the default value of the vertices' reposition bits or their presence in the stream as given in Table 10. When a default value is specified, the reposition bits are not present in the stream for the specified resolution level.

Table 10 — Encoding of the *repositionBits* flag

Value	Meaning
0	No reposition bits encoded, all have default value 0.
1	No reposition bits encoded, all have default value 1.
2	No default value, the reposition bits are present in the stream.
3	Reserved.

Further, if the *hasConnectivity* flag is set to '1' (the *hasConnectivityInfo* flag from the *MeshGridDecoderConfig* is set to '1') the **MeshDescriptor** class **(4)** computes the number of bits (*nBitsIndex*) in which the indices of the ROIs are stored, and **(5)** parses the number of bits (*nBitsConnectivity*) used for storing the length (in bytes) of the coded mesh connectivity. If the *hasReposition* flag is set to '1' and no default value is specified for the vertices' reposition bits (*repositionBits*), then **(6)** the number of bits (*nBitsReposition*) used for storing the length (in bytes) of the coded vertices' reposition information is retrieved. If the *hasRefine* flag is '1' (the *hasRefineInfo* flag from the *MeshGridDecoderConfig* is set to '1' and it is the last resolution level or the full refine flag has been specified), then **(7)** the number of bits (*nBitsRefine*) used for storing the length (in bytes) of the coded vertices' refinement information is parsed.

In subclause 5.2.2.21.1, Syntax for the RefineVertexDescriptor, replace the entire section with:

```
// number of refine bits vertex
class RefineVertexDescriptor {
    if (hasRefineInfo) {
        // full refine at each level
        bit(1)  bFull;

        // number of refine bits should be larger than 0
        // otherwise the hasRefineInfo flag should be 0
        unsigned int(REFINE_BITS) nBits;
    }
    else {
       unsigned int bFull = 0;
       unsigned int nBits = 0;
    }
}
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

Add subclause 5.2.2.32, MGDescriptor,

5.2.2.32 MGDescriptor
5.2.2.32.1 Syntax

```
class MGDescriptor extends MeshGridCommand: bit(8) tag=MGInfoTag
{
    // read the variable length counter sizeOfInstance
    MeshGridDecoderConfig decoderConfig;
}
```

5.2.2.32.2 Semantics

The MGDescriptor class parses the coding information of the MeshGrid stream. This part is mandatory to be present at the beginning of the stream when the MeshGrid stream is carried in the buffer field of the Bitwrapper node during the in-band scenario, as explained in ISO/IEC 14496-11.

In subclause 5.2.3.2.1, The 3D wavelet Decomposition, replace the entire section with:

The same 3D-wavelet decomposition is applied independently to each of the *x(u,v,w)*, *y(u,v,w)*, *z(u,v,w)* coordinates of the 3D reference grid.

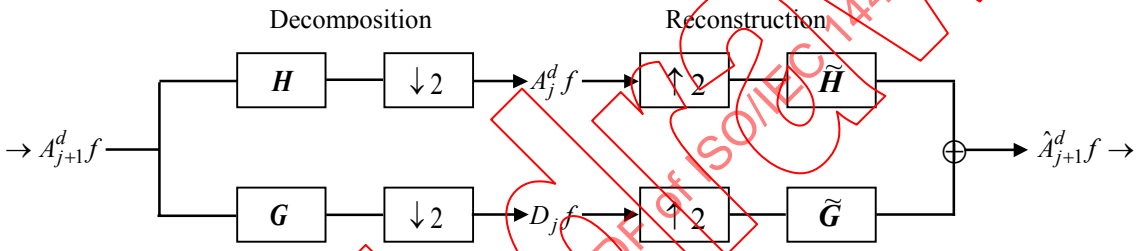


Figure 37 — Wavelet decomposition and reconstruction of a 1D signal.

For the 3D wavelet decomposition, the same analysis and synthesis 1D filters are used for each of the *u*, *v*, *w* directions. Two wavelet filters are supported, and the choice of the filter is specified by the *filterType* flag explained in section 5.2.2.2. Conform to the block scheme [62] shown in Figure 37, the two analysis low-pass and band-pass wavelet filters are respectively:

$$H(n) = \{1 \mid n = 0\},\ G(n) = \{-0.5, 1, -0.5 \mid n = -1,0,1\}, \text{ for } \textit{filterType} = 0, \qquad (12)$$

$$H(n) = \{1 \mid n = 0\},\ G(n) = \left\{\tfrac{1}{16}, 0, -\tfrac{9}{16}, 1, -\tfrac{9}{16}, 0, \tfrac{1}{16} \mid n = -3,-2,-1,0,1,2,3\right\}, \text{ for } \textit{filterType} = 1, \qquad (13)$$

The synthesis low-pass and band-pass wavelet filters are:

$$\tilde{H}(n) = \{0.5, 1, 0.5 \mid n = -1,0,1\},\ \tilde{G}(n) = \{1 \mid n = 0\}, \text{ for } \textit{filterType} = 0, \qquad (14)$$

$$\tilde{H}(n) = \left\{-\tfrac{1}{16}, 0, \tfrac{9}{16}, 1, \tfrac{9}{16}, 0, -\tfrac{1}{16} \mid n = -3,-2,-1,0,1,2,3\right\},\ \tilde{G}(n) = \{1 \mid n = 0\}, \text{ for } \textit{filterType} = 1. \qquad (15)$$

The wavelet filters are the same as the filters used for the hierarchical interpolation of the reference-grid points explained in subclause 4.3.4.3.1.

Further, in Figure 37, $A_j^d f(n)$ and $D_j f(n)$ are the discrete approximation and the detail signals respectively at the resolution $2^j, -L \le j \le -1$ of the input signal *f*, where *L* is the number of decomposition levels, and $A_0^d f(n) = f(n)$.

The low-pass filter H and the band-pass filter G are applied on the even and odd samples respectively of $A_{j+1}^d f(n)$ and the synthesis filters $\tilde{H}$ and $\tilde{G}$ are applied on the even and odd samples respectively of the low-pass and band-pass components $A_j^d f(n)$ and $D_j f(n)$.

The grid can be decoded at any resolution *j* if and only if the corners of the grid are stored in any $A_j^d f, -L \le j < 0$. There are some situations in which this constraint is not satisfied with the classical implementation of the pyramidal algorithm [62] discussed above. The wavelet analysis/synthesis filters given by (13) and (15) shall only be used when the constraint is satisfied for all decomposition levels $A_j^d f(n)$: the

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/Cor 1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

length of the discrete approximation signal $A_j^d f$ is odd. To solve this problem, a customized implementation of the pyramidal algorithm involves, in some situations, non-uniform down-sampling and up-sampling operations (see Figure 38(b,c)), coupled with analysis/synthesis filters that are different than the $H, G, \tilde{H}, \tilde{G}$ given by (12) and (14).

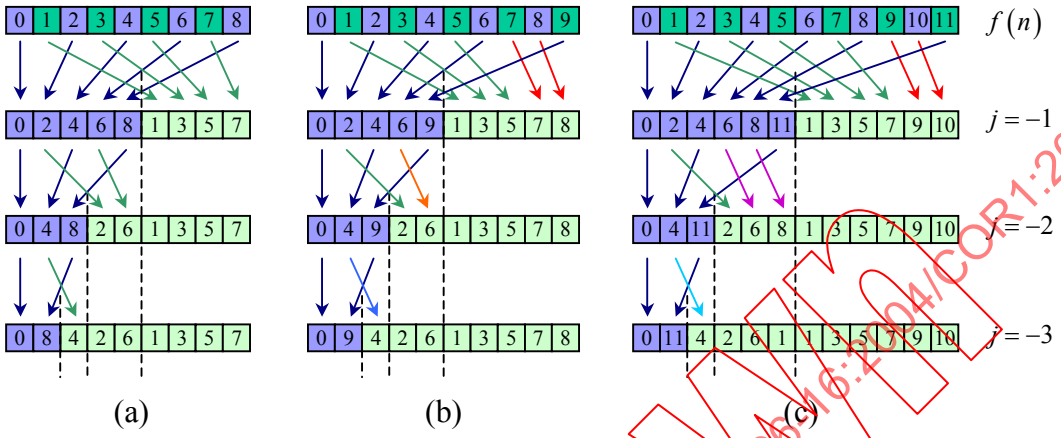


Figure 38 — Graphical illustration of the wavelet decomposition for odd and even-length signals.

As long as the length of the discrete approximation signal $A_j^d f$ is odd (see Figure 38 (a)), there is no need to perform a non-uniform down-sampling operation, and the classical pyramidal algorithm can be used. If at some resolution level $r, -L < r \le 0$ the length of $A_r^d f$ is even, then apart from the common situation in which G given by (14) is used, the analysis band-pass filters $G^1_{|r|+1}, G^2_{|r|+1}$ have to be used as well, and non-uniform down-sampling has to be applied for the last samples. This operation has to be repeated for all the resolution levels $p, -L < p \le r$. The filter coefficients of $G^1_{|p|+1}, G^2_{|p|+1}$ depend on the length of the discrete approximation signal $A_p^d f$ (whether it is an odd or an even number) and on the resolution level p. The additional filters used to derive the detail $D_{p-1} f$ starting from the discrete approximation $A_p^d f$ are:

$$G_p^1(n) = \{-c_2, 0, 1, -c_1 \mid n = -2, -1, 0, 1\},\ G_p^2(n) = \{-c_4, 1, 0, -c_3 \mid n = -1, 0, 1, 2\} \quad (16)$$

if the length of $A_p^d f$ is even, and:

$$G_p^1(n) = \{-c_6, 1, -c_5 \mid n = -1, 0, 1\} \quad (17)$$

if the length of $A_p^d f$ is odd. The constants $c_1, \ldots, c_6$ verify the relations:

$$c_1 = 1 - c_2,\ c_3 = 1 - c_4,\ c_4 = 2c_2, \text{ and } c_5 = 1 - c_6 \quad (18)$$

The constants c_2, c_6 are given by $c_2 = c_6 = x(n)/y(n)$, where $x(n)$ satisfies the recurrence $x(n) = x(n-1) + 2^{n-1}$, and $y(n)$ satisfies the recurrence $y(n) = y(n-1) + 2^{n-1}$ if the length of $A_p^d f$ is even, respectively $y(n) = y(n-1) + 2^n$ if the length of $A_p^d f$ is odd, with $x(0) = 1, y(0) = 3$ and $n = r - p$.

Add subclause 5.2.3.5, Decoding Quadrilateral MeshGrid:

5.2.3.5 Decoding Quadrilateral MeshGrid

According to the *sameBorderOrientation* flag from the *MeshGridDecoderConfig* class (subclause 5.2.2.2) quadrilateral meshes can be classified into: (1) type1 (generic) – *sameBorderOrientation* equals '0' – when the vertices may have different discrete border directions (see Figure 32), and (2) type2 – *sameBorderOrientation* equals '1' – when all the vertices have the same discrete border direction.

For the type2 quadrilateral MeshGrid the *uniformSplit* flag from the *MeshGridDecoderConfig* class can have the value '1', which means that any higher resolution level quadrilateral mesh can be obtained from a lower resolution level mesh by uniformly splitting each quad recursively into four sub-quads as illustrated in Figure 40. In addition if the *hasConnectivityInfo* flag from the *MeshGridDecoderConfig* class has the value '0' – meaning that there is no connectivity-wireframe stored in the stream – then a default type2 quadrilateral

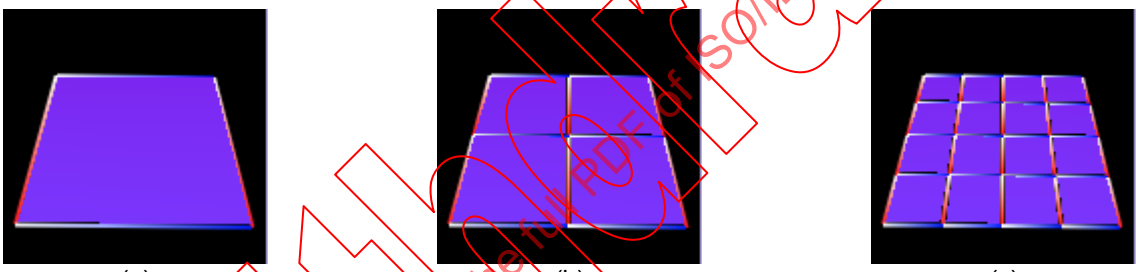
IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

connectivity-wireframe shall be constructed for the first resolution level. In this case the *uniformSplit* and *sameBorderOrientation* flags are set to '1'.

The reference-grid corresponding to the type2 quadrilateral MeshGrid with *hasConnectivityInfo* flag set '0' has the number of slices – *nSlices* from the *MeshGridDecoderConfig* class – in one of the directions {*u*,*v*,*w*} either equal to '2' – a double layer reference grid –, or equal to '1' – a single layer reference grid. For a double layer reference-grid the default type2 quadrilateral connectivity-wireframe shall be obtained by attaching a vertex to each reference-grid point belonging to the first layer of points i.e. the layer with the property that all the contained reference-grid points have one of the indices {*u*, *v*, *w*} equal to '0', and positioning these vertices on the reference-grid lines connecting the corresponding grid points from the first layer to the second layer. When the reference-grid is single layer, the default type2 quadrilateral connectivity-wireframe shall be obtained by attaching a vertex to each reference-grid point. The coordinates of the vertices can be computed as explained in subclause 4.3.4.3.2.

When the *hasRefineInfo* flag from the *MeshGridDecoderConfig* class (subclause 5.2.2.2) is set to '1' and there is a type2 quadrilateral MeshGrid with *hasConnectivityInfo* flag set '0', then the offsets (subclause 5.2.3.3) are specified only for the first resolution level of the mesh. The order of the offsets in the stream is the same as the indexing order of the reference-grid positions within the layer where the vertices are attached (the index corresponding to the layer is kept constant). The reference-grid points are indexed by incrementing two of the {*u*, *v*, *w*} indices in the order: *U* direction first, *V* direction second, and *W* direction third. The offset of each vertex corresponding to any higher resolution levels is computed as the average value of the offsets corresponding to the neighbour vertices' belonging to a lower resolution level. Namely, given V_1, V_2, V_3 and V_4 four vertices belonging to resolution level *l* (see Figure COR1.1), the offset of a vertex *V* belonging to level *l* + 1 located on a grid line of level *l* is computed as the average value of the offsets of V_1 and V_2, respectively the offset of vertex *V'* belonging to level *l* + 1 located on a grid line of level *l* + 1 is computed as the average value of the offsets of V_1, V_2, V_3 and V_4.



(a) (b) (c)

Figure COR1.1 — Example of a multi-resolution mesh with the connectivity-wireframe obtained by uniformly splitting each quad recursively into four sub-quads.

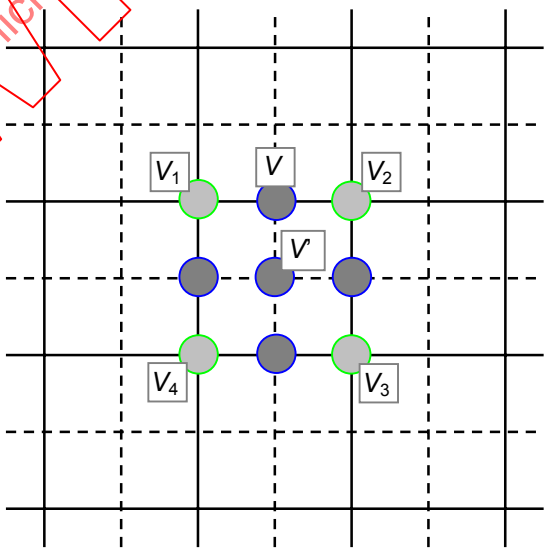


Figure COR1.2 — Computation of the offsets for the type2 quadrilateral MeshGrid.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/Cor 1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

In subclauses 5.5.3.13, 5.5.3.13.1, 5.5.3.14, 5.5.3.14.1, 5.5.3.15, 5.5.3.15.1, 5.5.3.16, 5.5.3.16.1, 5.5.3.17, 5.5.3.17.1, 5.5.3.18, 5.5.3.18.1, 5.5.3.19, 5.5.3.19.1, replace:

Commmand
with:
Command

In subclause 4.3.5, Particle systems, remove the entire section.

Add Clause 6, AFX object code:

6 AFX object code

AFXExtDescriptor described in ISO/IEC 14496-1 is an abstract class used as a placeholder for an optional DecoderSpecificInfo defined in Table COR1.2 :

Table COR1.2 — DecoderSpecificInfo for AFX streams.

AFX stream	DecoderSpecificInfo
MeshGrid	See subclause 5.2.2.2.
WaveletSubdivisionSurface	See subclause 5.1.1.1.
Other AFX streams	None

The tag field in the AFXExtDescriptor refers to a specific node compression scheme as defined in Table COR1.3.

Table COR1.3 — AFX object code.

AFX object code	Object
0x00	3D Mesh Compression
0x01	WaveletSubdivisionSurface
0x02	MeshGrid
0x03	CoordinateInterpolator
0x04	OrientationInterpolator
0x05	PositionInterpolator
0x06	OctreeImage
0x07	BBA

In A.1, Node coding tables, replace all the tables by the following set of tables:

Node Tables								
Node Name		**Node Data Type list**				**list id/context**		
Field name	***Field category***	***DEF id***	***IN id***	***OUT id***	***DYN id***	***[min, max]***	***Quantizer id***	***Animation method***
BitWrapper		SFWorldNode SF3DNode SF2DNode SFGeometryNode				000001 00001 0001 0001		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
node	SFWorldNode	00						
type	SFInt32	01						
url	MFURL	10						
buffer	SFString	11						

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

CoordinateInterpolator4D		SFWorldNode SF3DNode				000010 00010		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_fraction	SFFloat		00					
key	MFFloat	0	01	00		[0, 1]	8	
keyValue	MFVec4f	1	10	01		]-∞, +∞[	15	
value_changed	MFVec4f			10				

DepthImage		SFWorldNode SF3DNode SFDepthImageNode				000011 00011 1		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
diTexture	SFDepthTextureNode	000						
farPlane	SFFloat	001				[0, +∞[		
fieldOfView	SFVec2f	010				[0, □[		
nearPlane	SFFloat	011				[0, +∞[		
orientation	SFRotation	100						
orthographic	SFBool	101						
position	SFVec3f	110				]-∞, +∞[		

FFD		SFWorldNode SF3DNode				000100 00100		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
addChildren	MF3DNode		00					
removeChildren	MF3DNode		01					
children	MF3DNode	0000	10	0				
controlPoint	MFVec4f	0001	11	1	0	]-∞, +∞[	15	15
uDimension	SFInt32	0010				[2, 257]	13, 8	
uKnot	MFFloat	0011				]-∞, +∞[		
uOrder	SFInt32	0100				[2, 33]	13, 5	
vDimension	SFInt32	0101				[2, 257]	13, 8	
vKnot	MFFloat	0110				]-∞, +∞[		
vOrder	SFInt32	0111				[2, 33]	13, 5	
wDimension	SFInt32	1000				[2, 257]	13, 8	
wKnot	MFFloat	1001				]-∞, +∞[		
wOrder	SFInt32	1010				[2, 33]	13, 5	

Implicit		SFWorldNode SFGeometryNode				000101 0010		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
bboxSize	SFVec3f	000	000	000	0	[0, +∞[	11	11
c	MFFloat	001	001	001	1	]-∞, +∞[		7
densities	MFInt32	010	010	010		[0, +∞[		
dual	SFBool	011	011	011				
solid	SFBool	100	100	100				

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

MeshGrid		SFWorldNode SFGeometryNode				001100 0011		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_colorIndex	MFInt32		0000					
set_coordIndex	MFInt32		0001					
set_normalIndex	MFInt32		0010					
set_texCoordIndex	MFInt32		0011					
color	SFColorNode	00000	0100	0000				
coord	SFCoordinateNode	00001	0101	0001				
displayLevel	SFInt32	00010	0110	0010	000	[0, +∞[	13, 32	13
filterType	SFInt32	00011	0111	0011	001	[0, 1]	13, 2	13
gridCoord	SFCoordinateNode	00100	1000	0100				
hierarchicalLevel	SFInt32	00101	1001	0101	010	[-1, +∞[	13, 32	13
nLevels	MFInt32	00110	1010	0110	011		7	7
normal	SFNormalNode	00111	1011	0111				
nSlices	MFInt32	01000	1100	1000	100		7	7
texCoord	SFTextureCoordinateNode	01001	1101	1001				
vertexOffset	MFFloat	01010	1110	1010	101	[0, 2]	7	7
vertexLink	MFInt32	01011	1111	1011		[0, 3]	13, 2	
colorIndex	MFInt32	01100				[-1, +∞[	14	
coordIndex	MFInt32	01101				[-1, +∞[	14	
normalIndex	MFInt32	01110				[-1, +∞[	14	
solid	SFBool	01111						
texCoordIndex	MFInt32	10000				[-1, +∞[	14	
isLoading	SFBool			1100				
nVertices	MFInt32			1101				

NonLinearDeformer		SFWorldNode SFGeometryNode				001101 0100		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
axis	SFVec3f	000	000	000		[0, 1]		
extend	MFFloat	001	001	001				
geometry	SFGeometryNode	010	010	010				
param	SFFloat	011	011	011				
type	SFInt32	100	100	100		[0, 2]		

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

NurbsCurve		SFWorldNode SFGeometryNode				001110 0101		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_colorIndex	MFInt32		00					
color	SFColorNode	000	01	00				
controlPoint	MFVec4f	001	10	01	0	]-∞, +∞[	15	15
tessellation	SFInt32	010	11	10		[0, +∞[		
colorIndex	MFInt32	011					14	
colorPerVertex	SFBool	100						
knot	MFFloat	101				]-∞, +∞[		
order	SFInt32	110				[3, 34]	13, 5	

NurbsCurve2D		SFWorldNode SFGeometryNode				001111 0110		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_colorIndex	MFInt32		00					
color	SFColorNode	000	01	00				
controlPoint	MFVec3f	001	10	01	0	]-∞, +∞[	2	2
tessellation	SFInt32	010	11	10		[0, +∞[		
colorIndex	MFInt32	011					14	
colorPerVertex	SFBool	100						
knot	MFFloat	101				]-∞, +∞[		
order	SFInt32	110				[3, 34]	13, 5	

NurbsSurface		SFWorldNode SFGeometryNode				010000 0111		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_colorIndex	MFInt32		000					
set_texColorIndex	MFInt32		001					
color	SFColorNode	0000	010	000				
controlPoint	MFVec4f	0001	011	001	0	]-∞, +∞[	15	15
texCoord	SFTextureCoordinateNode	0010	100	010				
uTessellation	SFInt32	0011	101	011		[0, +∞[		
vTessellation	SFInt32	0100	110	100		[0, +∞[		
ccw	SFBool	0101						
colorIndex	MFInt32	0110					14	
colorPerVertex	SFBool	0111						
solid	SFBool	1000						
texColorIndex	MFInt32	1001					14	
uDimension	SFInt32	1010				[3, 258]	13, 8	
uKnot	MFFloat	1011				]-∞, +∞[		
uOrder	SFInt32	1100				[3, 34]	13, 5	

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

vDimension	SFInt32	1101				[3, 258]	13, 8	
vKnot	MFFloat	1110				]-∞, +∞[		
vOrder	SFInt32	1111				[3, 34]	13, 5	

OctreeImage		SFWorldNode SF3DNode				010001 00101		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
images	MFDepthImageNode	00						
octree	MFInt32	01				[0, 255]	13, 8	
octreeResolution	SFInt32	10				[1, +∞[		
voxelImageIndex	MFInt32	11				[0, 255]	13, 8	

PointTexture		SFWorldNode SFDepthTextureNode				010110 01		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
color	MFColor	000						
depth	MFInt32	001				[0, +∞[		
depthNbBits	SFInt32	010				[0, 31]	13, 5	
height	SFInt32	011				[1, +∞[		
width	SFInt32	100				[1, +∞[		

PositionAnimator		SFWorldNode SF3DNode				010111 00111		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_fraction	SFFloat		0000					
fromTo	SFVec2f	0000	0001	0000			8	
key	MFFloat	0001	0010	0001			8	
keyOrientation	MFRotation	0010	0011	0010				
keyType	SFInt32	0011	0100	0011				
keySpline	MFVec2f	0100	0101	0100			8	
keyValue	MFVec3f	0101	0110	0101			4	
keyValueType	SFInt32	0110	0111	0110				
offset	SFVec3f	0111	1000	0111		]-∞, +∞[	1	
weight	MFFloat	1000	1001	1000		[-1, 1]		
endValue	SFVec3f			1001				
rotation_changed	SFRotation			1010				
value_changed	SFVec3f			1011				

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

PositionAnimator2D		SFWorldNode SF2DNode SF3DNode				011000 0010 01000		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_fraction	SFFloat		0000					
fromTo	SFVec2f	0000	0001	0000			8	
key	MFFloat	0001	0010	0001			8	
keyOrientation	SFInt32	0010	0011	0010				
keyType	SFInt32	0011	0100	0011				
keySpline	MFVec2f	0100	0101	0100			8	
keyValue	MFVec2f	0101	0110	0101			4	
keyValueType	SFInt32	0110	0111	0110				
offset	SFVec2f	0111	1000	0111		]-∞, +∞[	2	
weight	MFFloat	1000	1001	1000		[-1, 1]		
endValue	SFVec2f			1001				
rotation_changed	SFFloat			1010				
value_changed	SFVec2f			1011				

PositionInterpolator4D		SFWorldNode SF3DNode				011001 01001		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_fraction	SFFloat		00					
key	MFFloat	0	01	00		[0, 1]	8	
keyValue	MFVec4f	1	10	01		]-∞, +∞[	15	
value_changed	SFVec4f			10				

ProceduralTexture		SFWorldNode SFTextureNode				011010 1		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
aSmooth	SFBool	00000	00000	00000				
aWarpmap	MFVec2f	00001	00001	00001		[0, 1]	2	
aWeights	MFFloat	00010	00010	00010		[-1, 1]		
bSmooth	SFBool	00011	00011	00011				
bWarpmap	MFVec2f	00100	00100	00100		[0, 1]	2	
bWeights	MFFloat	00101	00101	00101		[-1, 1]		
cellWidth	SFInt32	00110	00110	00110		[0, 15]	13, 4	
cellHeight	SFInt32	00111	00111	00111		[0, 15]	13, 4	
color	MFColor	01000	01000	01000			4	
distortion	SFFloat	01001	01001	01001		[0, 1]	13, 16	
height	SFInt32	01010	01010	01010		[1, 15]	13, 4	
roughness	SFInt32	01011	01011	01011		[0, 15]	13, 4	

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

seed	SFInt32	01100	01100	01100		]-∞, +∞[		
type	SFInt32	01101	01101	01101		[0, 4]	13, 3	
xSmooth	SFBool	01110	01110	01110				
xWarpmap	MFVec2f	01111	01111	01111		[0, 1]	2	
ySmooth	SFBool	10000	10000	10000				
yWarpmap	MFVec2f	10001	10001	10001		[0, 1]	2	
width	SFInt32	10010	10010	10010		[1, 15]	13, 4	
image_changed	SFImage			10011				

Quadric		SFWorldNode SFGeometryNode				011011 1000		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
bboxSize	SFVec3f	0000	0000	0000	000	[0, +∞[	11	11
densities	MFInt32	0001	0001	0001		[0, +∞[		
dual	SFBool	0010	0010	0010				
P0	SFVec4f	0011	0011	0011	001	]-∞, +∞[	15	15
P1	SFVec4f	0100	0100	0100	010	]-∞, +∞[	15	15
P2	SFVec4f	0101	0101	0101	011	]-∞, +∞[	15	15
P3	SFVec4f	0110	0110	0110	100	]-∞, +∞[	15	15
P4	SFVec4f	0111	0111	0111	101	]-∞, +∞[	15	15
P5	SFVec4f	1000	1000	1000	110	]-∞, +∞[	15	15
solid	SFBool	1001	1001	1001				

SBBone		SFWorldNode SF2DNode SF3DNode SFSBBoneNode				011100 0011 01010 1		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
addChildren	SF3DNode		00000					
removeChildren	SF3DNode		00001					
boneID	SFInt32	00000	00010	00000		[0, 1023]	13, 10	
center	SFVec3f	00001	00011	00001	000		1	1
children	MF3DNode	00010	00100	00010				
endpoint	SFVec3f	00011	00101	00011	001		1	1
falloff	SFInt32	00100	00110	00100		[-1, 4]	13, 3	
ikChainPosition	SFInt32	00101	00111	00101		[0, 3]	13, 2	
ikPitchLimit	MFFloat	00110	01000	00110				
ikRollLimit	MFFloat	00111	01001	00111				
ikTxLimit	MFFloat	01000	01010	01000				
ikTyLimit	MFFloat	01001	01011	01001				
ikTzLimit	MFFloat	01010	01100	01010				
ikYawLimit	MFFloat	01011	01101	01011				

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

rotation	SFRotation	01100	01110	01100	010		10	10
rotationOrder	SFInt32	01101	01111	01101		[0, 23]	13, 5	
scale	SFVec3f	01110	10000	01110	011		7	11
scaleOrientation	SFRotation	01111	10001	01111	100		10	10
sectionInner	MFFloat	10000	10010	10000				
sectionOuter	MFFloat	10001	10011	10001				
sectionPosition	MFFloat	10010	10100	10010				
skinCoordIndex	MFInt32	10011	10101	10011			14	
skinCoordWeight	MFFloat	10100	10110	10100		[-1, 1]		
translation	SFVec3f	10101	10111	10101	101		1	1

SBMuscle		SFWorldNode SF2DNode SF3DNode SFSBMuscleNode				011101 0100 01011 1		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
falloff	SFInt32	000	000	000		[-1, 4]	13, 3	
muscleCurve	SFGeometryNode	001	001	001				
muscleID	SFInt32	010	010	010		[0, 1023]	13, 10	
radius	SFInt32	011	011	011	0		7	11
skinCoordIndex	MFInt32	100	100	100		[0, +∞[	14	
skinCoordWeight	MFFloat	101	101	101		[-1, 1]		

SBSegment		SFWorldNode SF2DNode SF3DNode SFSBSegmentNode				011110 0101 01100 1		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
addChildren	MF3DNode		000					
removeChildren	MF3DNode		001					
centerOfMass	SFVec3f	000	010	000	0		1	1
children	MF3DNode	001	011	001				
mass	SFFloat	010	100	010				
momentsOfInertia	MFVec3f	011	101	011				
name	SFString	100	110	100				

SBSite		SFWorldNode SF2DNode SF3DNode SFSBSiteNode				011111 0110 01101 1		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
addChildren	MF3DNode		0000					
removeChildren	MF3DNode		0001					
center	SFVec3f	000	0010	000	000		1	1

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved

children	MF3DNode	001	0011	001				
name	SFString	010	0100	010				
rotation	SFRotation	011	0101	011	001		10	10
scale	SFVec3f	100	0110	100	010		7	11
scaleOrientation	SFRotation	101	0111	101	011		10	10
translation	SFVec3f	110	1000	110	100		1	1

SBSkinnedModel		SFWorldNode SF3DNode SF2DNode			100000 01110 0111			
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
bones	MFSBBoneNode	0000	0000	0000				
center	SFVec3f	0001	0001	0001	000		1	1
muscles	MFSBMuscleNode	0010	0010	0010				
name	SFString	0011	0011	0011				
rotation	SFRotation	0100	0100	0100	001		10	10
segments	MFSBSegmentNode	0101	0101	0101				
scale	SFVec3f	0110	0110	0110	010		7	11
scaleOrientation	SFRotation	0111	0111	0111	011		10	10
sites	MFSBSiteNode	1000	1000	1000				
skeleton	MF3DNode	1001	1001	1001				
skin	MF3DNode	1010	1010	1010				
skinCoord	SFCoordinateNode	1011	1011	1011				
skinNormal	SFNormalNode	1100	1100	1100				
translation	SFVec3f	1101	1101	1101	100		1	1
weighsComputationSkinCoord	SF3DNode	1110	1110	1110				

SBVCAnimation		SFWorldNode SF3DNode SF2DNode			100001 01111 1000		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
url	MFURL	0	0	0				
virtualCharacters	MF3DNode	1	1	1				

ScalarAnimator		SFWorldNode SF3DNode SF2DNode			100010 10000 1001		
Field name		***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
set_fraction	SFFloat		0000					
fromTo	SFVec2f	000	0001	0000			8	
key	MFFloat	001	0010	0001			8	
keyType	SFInt32	010	0011	0010				
keySpline	MFVec2f	011	0100	0011			8	
keyValue	MFFloat	100	0101	0100				
keyValueType	SFInt32	101	0110	0101				
offset	SFFloat	110	0111	0110				

IECNORM.COM : Click to view the full PDF of ISO/IEC 14496-16:2004/COR1:2005
Withdrawn

© ISO/IEC 2005 – All rights reserved