INTERNATIONAL STANDARD

ISO/IEC 14496-16

Fourth edition
2011-11-01

AMENDMENT 4
2017-10

Information technology — Coding of audio-visual objects —

Part 16: Animation Framework eXtension (AFX)

AMENDMENT 4: Pattern-based 3D mesh coding (PB3DMC)

Technologies de l'information — Codage des objets audiovisuels —

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

AMENDEMENT 4: Codage de maille 3D fondé sur un modèle

Reference number
ISO/IEC 14496-16:2011/Amd.4:2017(E)

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

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

A list of all parts in the ISO/IEC 14496 series can be found on the ISO website.

Information technology — Coding of audio-visual objects —

Part 16: Animation Framework eXtension (AFX)

AMENDMENT 4: Pattern-based 3D mesh coding (PB3DMC)

5.2 Geometry tools

Add the following subclauses:

5.2.7 Pattern-based 3D mesh coding (PB3DMC)

5.2.7.1 Overview

In practical applications, many 3D models consist of a large number of connected components. And these multi-connected 3D models usually contain lots of repetitive structures in various transformations, as shown in Figure Amd4.1. In order to increase their efficiency, compression methods for this kind of 3D models should be able to extract the redundancy existing in the repetitive structures.

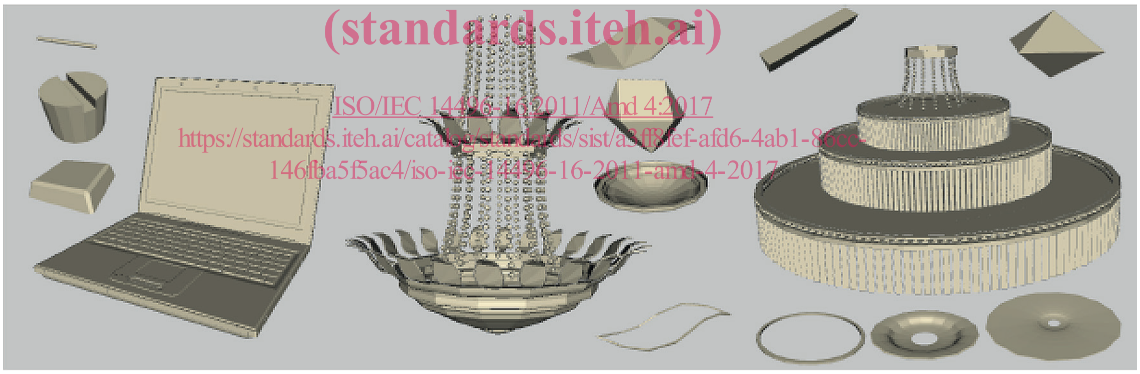


Figure Amd4.1 — 3D models with a large number of connected components and repetitive structures

This document presents an efficient compression algorithm for multi-connected 3D models by taking advantage of discovering repetitive structures in the input models. It allows discovering of structures repeating in various positions, orientations and scaling factors, where the 3D model is then organized into "pattern-instance" representation. A pattern is the representative geometry of the corresponding repetitive structure. The connected components belonging to a repetitive structure are called instances of the corresponding pattern and represented by the pattern ID and their transformation, i.e. the combination of reflection, translation, rotation and possible uniform scaling, with regards to the pattern. The instance transformation consists of four parts: reflection part, translation part, rotation part and possible scaling part.

The repetitive structure discovery-based compression algorithm proposed in this subclause, brings significant compression gain compared to the static 3D model compression algorithms provided by SC3DMC when 3D models present repetitive features. This document defines the compressed bitstream syntax and semantics for this repetitive structure discovery-based compression approach.

5.2.7.2 General compression approach

5.2.7.2.1 Identification of repetitive and symmetric structures in a 3D model

This subclause defines several terms that are used all across this document and presents the different steps that are used in order to identify repetitive structures.

There are two types of repetitive structures in 3D models:

— ***Unconnected repetitive structure***: One unconnected repetitive structure consists of all the connected components which are invariant in various positions, orientations and scaling factors. One unconnected repetitive structure includes one pattern, which corresponds to one connected component in this structure, and all the other connected components are its instances.

— ***Connected repetitive structure***: One connected repetitive structure consists of all the surface patches which are invariant in various positions, orientations and scaling factors. In other words, it is a repetitive structure that may be found within one connected component. Connected repetitive structure is sometimes also called ***symmetric structure***. This document uses both naming. Similar to unconnected repetitive structure, one symmetric structure includes one pattern, which corresponds to one surface patch in this structure, and all the other surface patches are instances of the pattern.

PB3DMC discovery of the two types of repetitive structures is done in four steps as described below, considering that the input 3D model is as shown in Figure Amd4.2.

Figure Amd4.2 — Input 3D model

STEP 1: Identification of unconnected repetitive structures and unique part.

The input 3D model is divided into two parts, the unconnected repetitive structures and the unique part which includes all those components that are not included in any unconnected repetitive structures. In Figure Amd4.2, clouds, leaves and houses on the right are all unconnected repetitive structures.

STEP 2: Choose patterns and instances within unconnected repetitive structures.

Following STEP 1, for each unconnected repetitive structures, one pattern shall be chosen among all repetitions. Other repetitions then become instances of their related pattern. The input 3D model is then divided into three parts as follows:

— A: Patterns of all unconnected repetitive structures (see Figure Amd4.3);

— B: Instances of all unconnected repetitive structures (see Figure Amd4.4);

— C: Unique part which is not included in any unconnected repetitive structures (see Figure Amd4.5).

Figure Amd4.3 — A: Patterns of all unconnected repetitive structures

Figure Amd4.4 — B: Instances of all unconnected repetitive structures

Figure Amd4.5 — C: Unique part which is not included in any unconnected repetitive structures

As shown in Figure Amd4.6, part A can be further divided into two types:

— D: Patterns of unconnected repetitive structures which do not include any symmetric structures;

— E: Patterns of unconnected repetitive structures which include symmetric structures.

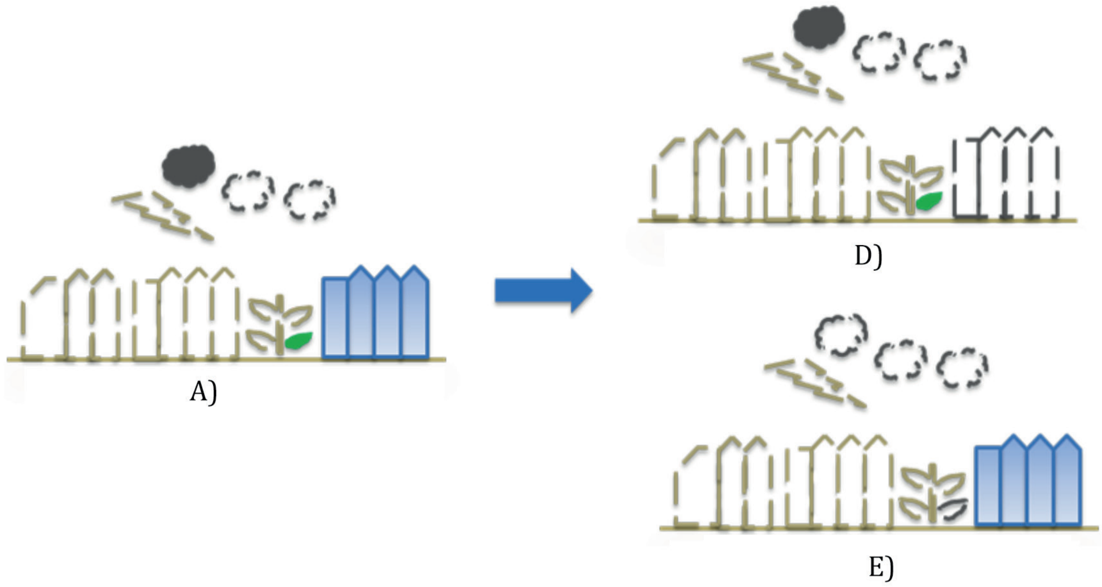


Figure Amd4.6 — Part A further divided into two types

As shown in Figure Amd4.7, part C can be further divided into two types:

— F: Unique part which is not included in any unconnected repetitive structures and which does not include unconnected repetitive structures and symmetric structures;

— G: Unique part which is not included in any unconnected repetitive structures and which includes symmetric structures.

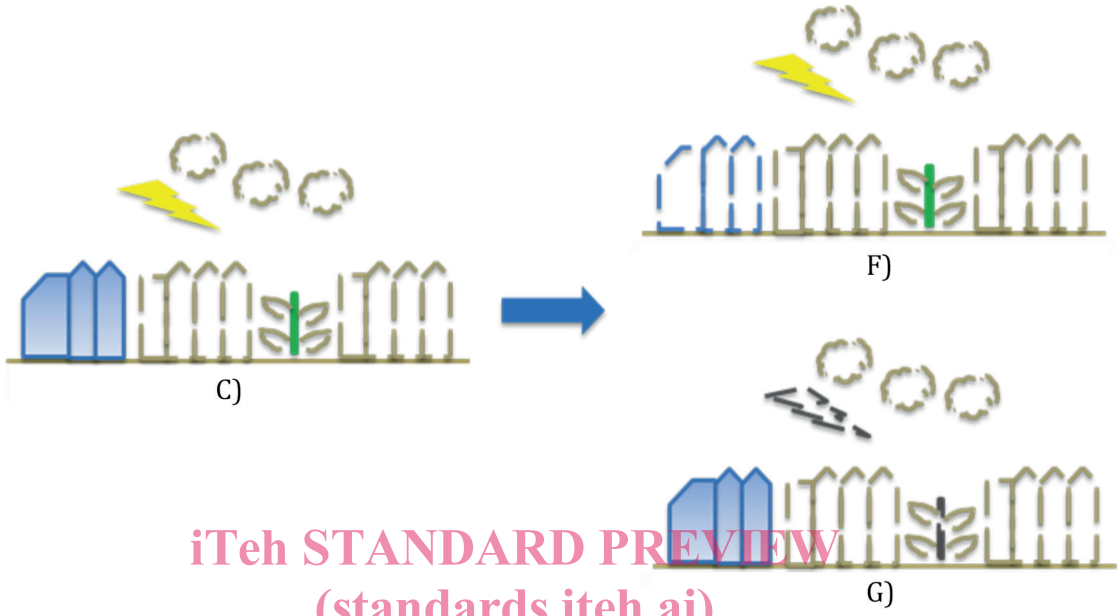


Figure Amd4.7 — Part C can be further divided into two types

STEP 3: Identification of symmetric structures.

Since instances will be coded with reference to their related patterns, symmetric structures are identified either within pattern or unique parts in the 3D model. Consequently, as shown in Figure Amd4.8, the input in STEP 3 is the input model, except instances of all unconnected repetitive structures discovered in STEP 2.

NOTE The input in STEP 3 is the input model, except instances of unconnected repetitive structure.

Figure Amd4.8 — Input of STEP 3

STEP 4: Choose patterns and instances within symmetric structures.

Similar to STEP 2 on repetitive structures, STEP 4 consists choosing patterns and instances within symmetric structures. As shown in Figure Amd4.9, the following four types of data are defined:

— H: Patterns of all symmetric structures;

— I: Instances of all symmetric structures;

— J: Unique parts of those unique components which do not belong to any unconnected repetitive structures but have symmetric structures;

— K: Unique parts on those unconnected-repetitive-structure patterns including symmetric structures.

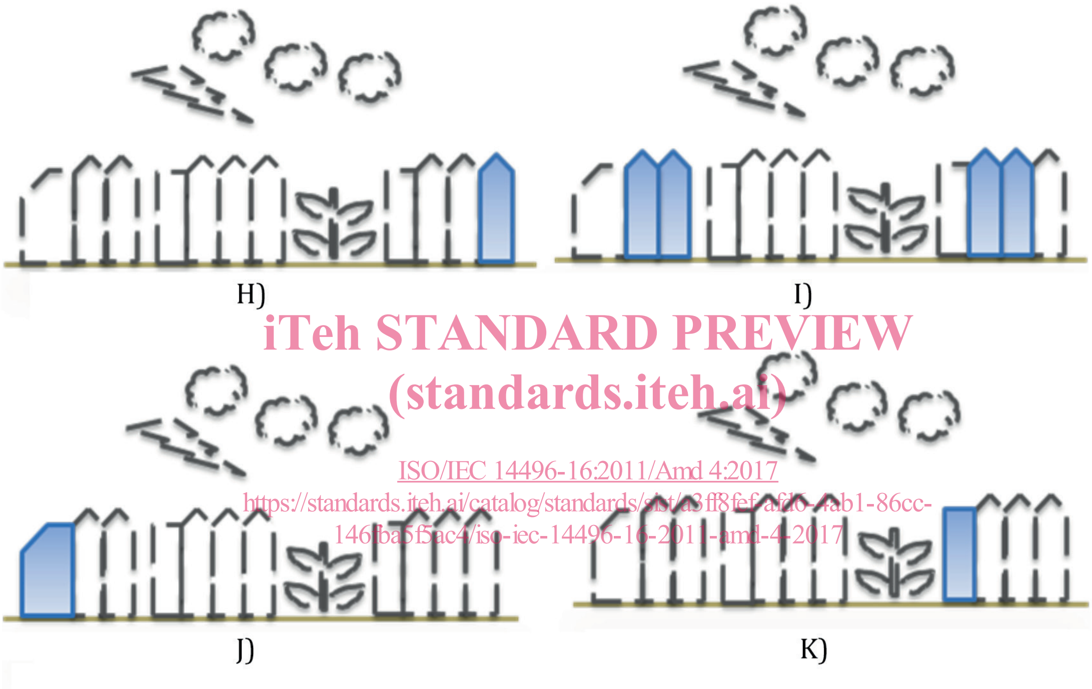


Figure Amd4.9 — Result of STEP 4

5.2.7.2.2 Two-instance reconstruction modes

While the bitstream needs to embed all the instance data, it should also be efficient and should address several applications where sometimes either bitstream size or decoding efficiency or error resilience matters the most.

Therefore, two options are proposed in reconstructing the data of one instance, i.e. its pattern ID (ID being the actual position of the patterns in the compressed bitstream of patterns: 1 for first pattern, 2 for second pattern, etc.), its reflection transformation part (F), its translation transformation part (T), its rotation transformation part (R) and its scaling transformation part (S), from the bitstream. Both of them have their own pros and cons.

Option (A) elementary instance data mode (ID, F, T, R, S, ID, F, T, R, S...): Using this mode, the pattern ID, reflection transformation part, translation transformation part, rotation transformation part and scaling transformation part of one instance are packed together in the bitstream.

Pros:

— It is error resilient. The decoder can recover from losing the transformation of some instances.

— Online decoding, which means that the instances can be decoded one by one during actual reading of the compressed bitstream. There is no need to wait for the reading of the whole compressed bitstream to be finished.

— Higher codec speed.

— The codec needs no buffer.

Cons:

— Relatively larger compressed 3D model size.

Option (B) grouped instance data mode (ID, ID, F, F, T, T, R, R, S, S): Using this mode, the pattern ID, reflection transformation part, translation transformation part, rotation transformation part and scaling transformation part of one instance are packed together in the bitstream.

Pros:

— Relatively smaller compressed 3D model size.

Cons:

— The decoder is no longer error resilient.

— Off-line decoding, which means the decoder can only start decoding after reading the whole compressed bitstream.

— Lower codec speed.

— Buffer is necessary.

The bitstream definition includes both of the above two options. The encoder can choose the one which fits its application better.

Since instances may have larger reconstruction error than its related patterns (error being defined as the distance between the original component and the component restored from the pattern and instance transformation), some data fields of the bitstream are defined to denote the compressed instance reconstruction error to guarantee the decoded 3D model quality. Whether or not to record the decoding error of an instance is based on the quality requirement.

5.2.7.2.3 Reconstruction of instance transformation

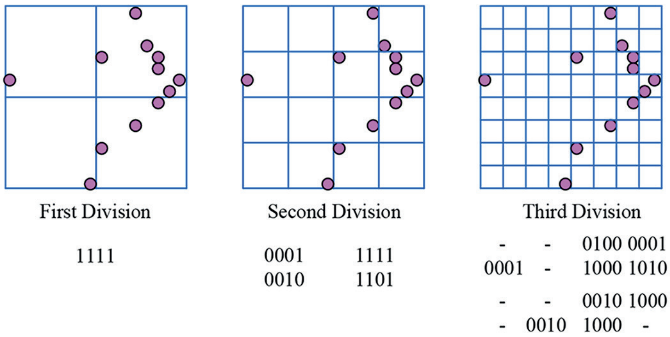
The instance transformation is reconstructed from four parts: reflection part, rotation part, translation part and possible scaling part.

— The reflection part is represented by a 1-bit flag.

— The rotation part is reconstructed from the three Euler angles (alpha, beta, gamma).

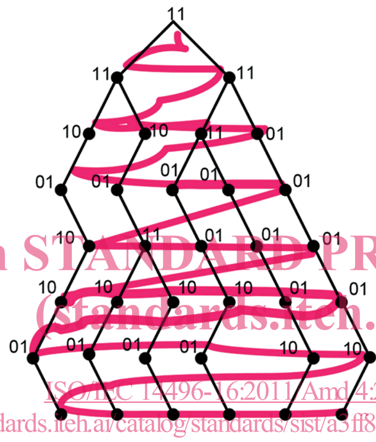
— The translation part is represented by a vector (x, y, z) (translation vector).

— The scaling part is represented by the uniform scaling factor S of the instance.



NOTE The number under each sub-figure is the corresponding occupancy codes.

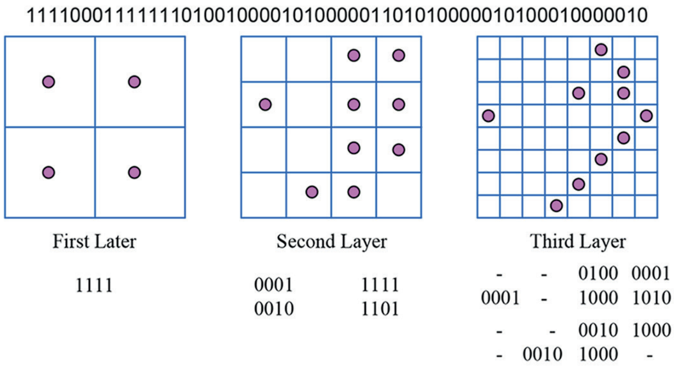
a) 2D example of space subdivision for quad-tree construction



NOTE 1 The red line illustrates the breadth first traversal of the binary tree.

NOTE 2 The occupancy codes of the tree nodes, sorted according to the breadth first traversal, construct the occupancy code sequence that describes the tree structure.

b) Example of breadth first traversal of the binary tree



NOTE For a predefined traversal order “top left–top right–bottom right–bottom left”, the input bitstream is decoded.

c) 2D example of quad-tree reconstruction process

Figure Amd4.10 — Reconstruction of instance transformation

While using grouped instance transformation mode, all instance translation vectors are compressed by octree (OT) decomposition-based compression algorithm, which recursively subdivides the bounding box of all instance translation vectors in an octree data structure, as illustrated by the 2D example in Figure Amd4.10 a). Each octree node subdivision is represented by the 8-bit long occupancy code, which uses a 1-bit flag to signify whether a child node is nonempty. An occupancy code sequence describing the octree is generated by breadth first traversing the octree, as illustrated by the 2D example in Figure Amd4.10 b). To decode these instance translation vectors, the octree is reconstructed by breadth first traversing the octree from top to bottom, as shown in Figure Amd4.10 c). First, the top layer is obtained by decoding the first long occupancy code. If any nodes at Layer i have more than one "1" in its occupancy code, e.g. "01100000", the codec decodes necessary number of symbols to append as the children of such nodes. This process continues until all the leaf nodes have only one "1" or are the terminal code. The occupancy code sequence is decoded by dividing it into several intervals and decoded them with different probability models. Since instances may have extremely close translation vectors, which are called as duplicate translation vectors, some data fields of the bitstream are defined to denote these duplicate translation vectors.

5.2.7.2.4 Reconstruction of instance attributes

In practical applications, besides geometry, 3D models usually have various attributes such as normal, colour and texture coordinates. Requiring instances to have the same attributes of patterns will limit the number of repetitive structures that can be discovered and decrease the compression ratio of PB3DMC. Thus, only the geometry is checked during repetitive structure discovery and the instances may have attributes different from the corresponding pattern's attributes. There are two reconstruction modes for instance attributes.

— Share attribute mode: The instance shares the pattern attribute data and does not need data fields to represent its attributes.

— Specific attribute mode: The instance has its own attributes and need separate data fields to represent its attributes in the bitstream.

When the elementary instance data mode is used, one data field is defined to denote how to reconstruct the attributes of an instance from the bitstream. The attribute data of one instance (A) follows the other data of the instance, i.e. (ID, F, T, R, S, A, ID, F, T, R, S, A…). When the grouped instance data mode is used, all instances should either share the pattern attribute data or have their own attribute data. The instance data part of the bitstream is like (ID, ID, F, F, T, T, R, R, S, S, A, A).

5.2.7.2.5 Reconstruction of textured image(s)

It is expected that the repetitive structures in 3D models share textured image(s)/portion(s), as well as geometry, as shown in Figure Amd4.11. Given the geometric matching relationship between patterns and instances, the texture redundancy can be removed. Thus, the compression ratio can be further improved by removing the textured image redundancy with the help of geometric repetition information.