International Standard

ISO/IEC 23090-29

Information technology — Coded representation of immersive media —

Part 29:
Video-based dynamic mesh coding (V-DMC)

Technologies de l'information — Représentation codée de média immersifs —

Partie 29: Codage dynamique de maillages basé vidéo (V-DMC)

First edition
2026-07

Reference number
ISO/IEC 23090-29:2026(en)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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 ISO documents 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 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 of 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 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 23090 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

The advances in 3D capture, modelling, and rendering have promoted the ubiquitous presence of 3D content across several platforms and devices.

3D meshes, along with point clouds and immersive video, are widely used to represent such immersive content. A mesh is composed of several polygons that describe the boundary surface of a volumetric object. Each polygon is defined by its vertices in 3D space and the information on how the vertices are connected, referred to as connectivity information. Optionally, vertex attributes, such as colours, normal, etc., can be associated with the mesh vertices. Attributes can also be associated with the surface of the mesh by exploiting mapping information that describes a parameterization of the mesh onto 2D regions of the plane. Such mapping is usually described by a set of parametric coordinates, referred to as UV coordinates or texture coordinates. 2D attribute maps are used to store high resolution attribute information such as texture, normal, material ID etc. Such information can be used for various purposes such as texture mapping and shading.

To achieve a realistic representation of an object or scene, 3D meshes are becoming increasingly sophisticated, and a significant amount of data is generated for the creation and consumption of those meshes. When uncompressed, these data can be costly in terms of storage and transmission. Furthermore, dynamic mesh sequences may require even larger amounts of data, since it may consist of a significant quantity of information changing over time. A visual volumetric video-based coding (V3C) standard has been created to support the distribution of volumetric data, currently addressing point clouds (ISO/IEC 23090-5) and immersive video content (ISO/IEC 23090-12). Therefore, this document was developed to support compression of 3D video-based mesh coding utilizing the ISO/IEC V3C standard (ISO/IEC 23090-5).

Information technology — Coded representation of immersive media —

Part 29: Video-based dynamic mesh coding (V-DMC)

1 Scope

This document specifies syntax, semantics, and decoding for video based dynamic mesh coding (V-DMC), basemesh coding, MPEG edgebreaker static mesh coding, and arithmetic coded displacement. Furthermore, this document specifies processes that can be used for reconstruction of visual volumetric media and also include additional processes such as post-decoding, pre-reconstruction, post-reconstruction, and adaptation.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO /IEC 23090-5:2026, *Information technology — Coded Representation of Immersive Media — Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23090-5 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
atlas
collection of 2D bounding boxes and their associated information placed onto a rectangular frame and corresponding to a volume in 3D space on which volumetric data is rendered and a list of metadata corresponding to a part of a surface of a mesh in 3D space

3.2
atlas frame
2D rectangular array of atlas samples onto which patches are projected and additional information related to the patches, corresponding to a volumetric frame and a list of *meshpatches* (3.8) and additional information related to the *meshpatches* (3.8), corresponding to a volumetric frame

3.3
atlas sample
position on the rectangular frame onto which patches that are associated with an *atlas* (3.1) are projected or element of a list of *meshpatches* (3.8) that are associated with an *atlas* (3.1)

3.4
attribute frame
2D rectangular array created through the aggregation of patches and *meshpatches* (3.8) containing values of a specific attribute

3.5
attribute map
image for mapping an attribute into a surface of 3D shape

3.6
basemesh frame
array of *submeshes* (3.12) associated with each *meshpatch* (3.8) corresponding to a volumetric frame

3.7
geometry frame
2D array created through the aggregation of the geometry information associated with each patch and *meshpatch* (3.8)

3.8
meshpatch
element of an *atlas* (3.1) associated with basemesh information

3.9
meshpatch data
data in an *atlas* (3.1) associated with a *meshpatch* (3.8) that enables the conversion of basemesh into the reconstructed mesh

3.10
orthoAtlas
orthographic projection and packing of the faces into a texture atlas

3.11
subdivision
process of dividing the mesh faces into a number of sub-faces

3.12
submesh
independently decodable region of a basemesh

3.13
subpatch
set of parameters in the *meshpatch* (3.8) to construct a homography transform used to map the vertex coordinates of a mesh surface associated with a given faceId to a set of UV texture coordinates and their connectivity indices

3.14
texture mapping
process of transferring attributes from a 2D attribute (texture) map into surface of a 3D mesh

3.15
atlas tile
independently decodable rectangular region of *atlas frame* (3.2), consisting of a *geometry tile* (3.16), an *attribute tile* (3.17), or both, with no dependencies on other atlas tiles in the current frame

3.16
geometry tile
independently decodable rectangular region within an *atlas tile* (3.15) that is associated with a *geometry frame* (3.7)

3.17
attribute tile
independently decodable rectangular region within an *atlas tile* (3.15) that is associated with an *attribute frame* (3.4)

3.18
basemesh layer set
set of layers represented within a bitstream created from another bitstream by operation of the *basemeshsub-bitstream extraction process* (3.19) with another bitstream, and the target highest BmTemporalId equal to 6 as inputs

3.19
basemesh sub-bitstream extraction process
specified process by which basemesh NAL units in a bitstream that do not belong to a target set, determined by a target highest BmTemporalId tBmIdTarget, are removed from the bitstream, with the output sub-bitstream consisting of the basemesh NAL units in the bitstream that belong to the target set. This process removes all NAL units with BmTemporalId greater than tBmIdTarget

3.20
basemesh sub-layer
temporal basemesh sub-layer
temporal scalable layer of a temporal scalable basemesh sub-bitstream, consisting of BMCL NAL units with a particular value of the BmTemporalId variable and the associated non-BMCL NAL units

3.21
coded basemesh
coded basemesh frame
coded representation of a basemesh frame

3.22
coded basemesh access unit
set of basemesh NAL units that are associated with each other according to a specified classification rule, are consecutive in decoding order, and contain all basemesh NAL units pertaining to one particular output time

3.23
decoded basemesh
decoded basemesh frame
basemesh derived by decoding a *coded basemesh* (3.21)

3.24
motion decoder
process used for inter basemesh decoding

3.25
motion sub-bitstream
sequence of bits extracted from V3C bitstream that contains specified syntax that represents motion of a basemesh

3.26
operation point
bitstream created from another bitstream by operation of the *basemeshsub-bitstream extraction process* (3.29) with another bitstream, and a target highest BmTemporalId as inputs

3.27
pruned decoded basemesh
result of pruning a *decoded basemesh* (3.23) by removing member from the struct *decoded basemesh* (3.23) that are not specified at the decoding output

3.28
sub-layer non-reference basemesh frame
SLNR basemesh frame
basemesh *frame* (3.6) that cannot be used for inter prediction in the decoding process of subsequent basemesh frames of the same *basemesh sub-layer* (3.20) in decoding order

Note 1 to entry: An SLNR basemesh frame can be used for inter prediction in the decoding process of subsequent basemesh frames of higher basemesh sub-layers in decoding order.

3.29
sub-layer reference basemesh frame
basemesh frame (3.6) that may be used for inter prediction in the decoding process of subsequent basemesh frames of the same *basemesh sub-layer* (3.20) in decoding order

Note 1 to entry: A sub-layer reference basemesh frame can also be used for inter prediction in the decoding process of subsequent basemesh frames of higher basemesh sub-layers in decoding order.

3.30
sub-layer representation
subset of the bitstream consisting of basemesh NAL units of a particular *basemesh sub-layer* (3.20) and the lower basemesh sub-layers

3.31
step-wise temporal sub-layer access basemesh/STSA basemesh
coded basemesh for which each BMCL NAL unit has bmesh_nal_unit_type equal to STSA_R or STSA_N

Note 1 to entry: An STSA basemesh does not use basemeshes with the same BmTemporalId as the STSA basemesh for inter prediction reference. Basemeshes following an STSA basemesh in decoding order with the same BmTemporalId as the STSA basemesh do not use basemeshes prior to the STSA basemesh in decoding order with the same BmTemporalId as the STSA basemesh for inter prediction reference. An STSA basemesh enables up-switching, at the STSA basemesh, to the basemesh sub-layer containing the STSA basemesh, from the immediately lower basemesh sub-layer. STSA basemeshes have BmTemporalId greater than 0.

3.32
temporal sub-layer access unit/TSA access unit
access unit in which the *coded basemesh* (3.21) with bmesh_nal_layer_id equal to 0 is a TSA basemesh

3.33
temporal sub-layer access basemesh/TSA basemesh
coded basemesh for which each BMCL NAL unit has bmesh_nal_unit_type equal to TSA_R or TSA_N

Note 1 to entry: A TSA basemesh and basemeshes following the TSA basemesh in decoding order do not use basemeshes prior to the TSA basemesh in decoding order with BmTemporalId greater than or equal to that of the TSA basemesh for inter prediction reference. A TSA basemesh enables up-switching, at the TSA basemesh, to the sub-layer containing the TSA basemesh or any higher sub-layer, from the immediately lower sub-layer. TSA basemeshes have BmTemporalId greater than 0.

3.34
attribute array
attribute table
collection of attribute values accessible with a unique index associated with each attribute vertex

3.35
attribute seam
boundary edge (3.39) associated with the *auxiliary connectivity* (3.37) that is not a *boundary edge* (3.39) on the *primary connectivity* (3.58)

3.36
auxiliary attribute
set of values attached to either each attribute vertex, or each face of the *mesh* (3.54), accessed by an index in an *attribute table* (3.34)

3.37
auxiliary connectivity
connectivity (3.42) associated with the *auxiliary attribute* (3.36), which can be identical to the *primary connectivity* (3.58)

3.38
boundary
closed loop connecting *boundary edges* (3.39)

3.39
boundary edge
edge incident to only one face

3.40
CLERS
set of symbols used to encode the *primary connectivity* (3.58)

3.41
connected component
maximal connected subgraph of the *connectivity* (3.42)

3.42
connectivity
graph defining face to vertex incidence

3.43
corner
incidence of a vertex and a face

3.44
corner table
table representing the *connectivity* (3.42) by associating each *corner* (3.43) to both a vertex and an opposite *corner* (3.43), where *boundary edges* (3.39) are signaled by negative opposite *corner* (3.43) values

3.45
degenerate face
face that dereferences to the same index in an *attribute table* (3.34) for at least two of its *corners* (3.43)

3.46
external boundary
first *boundary* (3.38) in a *connected component* (3.41) traversed when encoding its *connectivity* (3.42)

3.47
fan
set of connected faces that share a unique central vertex, where each face in the set shares at least one edge with another face in the set when the fan consists of more than one face, and where each edge is shared by at most two faces

3.48
handle
topological feature distinguishing a torus from a sphere, where the number of such topological features in a *manifold mesh* (3.53) with *boundaries* (3.38) is identified by the Euler-Poincaré characteristic

3.49
handle table
table containing a list of pairs of values representing opposite corners as hints for the reconstruction of the correct surface *connectivity* (3.42) with *handles* (3.48) or *internal boundaries* (3.51) during decoding

3.50
indexed face set
structure containing *attribute arrays* (3.34), and arrays associating each corner of the *mesh* with an index dereferencing those for each of the *primary attribute* (3.57) and the *auxiliary attributes* (3.36)

3.51
internal boundary
boundary (3.38) attached to a *connected component* (3.41) with more than one *boundary* (3.38), and which is not the *external boundary* (3.46) of this *connected component* (3.41)

3.52
isolated vertices
vertex that is not connected to any other vertex

3.53
manifold mesh
mesh (3.54) for which each edge is incident to only one or two faces, and the set of faces incident to a vertex form a closed or an open *fan* (3.47)

3.54
mesh
collection of faces that are connected by their per attribute common edges and vertices, with a per attribute cyclic order of *corners* (3.43) that defines the per-attribute orientation associated with each face

3.55
orientable manifold mesh
manifold mesh (3.53) in which for all per vertex attributes any two adjacent faces have compatible orientation, if the two vertices of the common edge are in opposite order in the cyclic order of each respective face

3.56
orientable manifold mesh with boundaries
orientable manifold mesh (3.55) that has one or more *boundaries* (3.38)

3.57
primary attribute
first set of values attached to each vertex of a *mesh* (3.54), accessed by an index in an *attribute table* (3.36)

3.58
primary connectivity
connectivity (3.42) associated with the *primary attribute* (3.57)

3.59
triangular mesh
mesh (3.54) where faces correspond to triangles

3.60
subdisplacement
independently decodable *displacement* (3.8) data associated with a *submesh* (3.12)

3.61
coded basemesh sequence
CBMS
sequence of *coded basemesh access units* (3.22), in decoding order, of an *IRAP coded basemesh access unit* (3.62) with NoBmOutputBeforeRecoveryFlag equal to 1, followed by zero or more *coded basemesh access units* (3.22) that are not *IRAP coded basemesh access units* (3.62) with NoBmOutputBeforeRecoveryFlag equal to 1, including all subsequent *coded basemesh access units* (3.22) up to but not including any subsequent *coded basemesh access unit* (3.22) that is an *IRAP coded basemesh access unit* (3.62) with NoBmOutputBeforeRecoveryFlag equal to 1

3.62
intra random access point coded basemesh access unit
IRAP coded basemesh access unit
access unit in which the *coded basemesh* (3.21) with bmesh_nal_layer_id equal to 0 is an *IRAP coded basemesh frame* (3.63)

3.63
intra random access point coded basemesh frame
IRAP coded basemesh frame
coded basemesh (3.21) for which each *BMCL NAL unit* (3.66) has bmesh_nal_unit_type in the range of BNAL_BLA_W_LP to BNAL_RSV_IRAP_BMCL_23, inclusive

3.64
basemesh NAL unit
BNAL unit
syntax structure within a basemesh bitstream containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP

3.65
coded submesh NAL unit
basemesh NAL unit (3.64) that has bmesh_nal_unit_type in the range of BNAL_TRAIL_N to BNAL_SKIP_R, inclusive, or in the range of BNAL_BLA_W_LP to BNAL_RSV_IRAP_BMCL_23, inclusive, which indicates that the *basemesh NAL unit* (3.64) contains a coded *submesh* (3.12)

3.66
basemesh coding layer NAL unit
BMCL NAL unit
collective term for *coded submesh NAL units* (3.65) and the subset of *basemesh NAL units* (3.64) that have reserved values of bmesh_nal_unit_type that are classified as BMCL NAL units

3.67
non-BMCL NAL unit
basemesh NAL unit (3.64) that is not an *BMCL NAL unit* (3.66)

3.68
coded displacement sequence
CDS
sequence of *coded displacement access units* (3.71), in decoding order, of an *IRAP coded displacement access unit* (3.69) with NoDisplOutputBeforeRecoveryFlag equal to 1, followed by zero or more *coded displacement access units* (3.71) that are not *IRAP coded displacement access units* (3.69) with NoDisplOutputBeforeRecoveryFlag equal to 1, including all subsequent *coded displacement access units* (3.71) up to but not including any subsequent *coded displacement access unit* (3.71) that is an *IRAP coded displacement access unit* (3.69) with NoDisplOutputBeforeRecoveryFlag equal to 1

3.69
intra random access point coded displacement access unit
IRAP coded displacement access unit
access unit in which the coded displacement with displ_nal_layer_id equal to 0 is an *IRAP coded displacement frame* (3.70)

3.70
intra random access point coded displacement frame
IRAP coded displacement frame
coded displacement for which each *DCL NAL unit* (3.76) has displ_nal_unit_type in the range of DNAL_BLA_W_LP to DNAL_RSV_IRAP_DCL_23, inclusive

3.71
coded displacement access unit
set of *displacement NAL units* (3.74) that are associated with each other according to a specified classification rule, are consecutive in decoding order, and contain all *displacement NAL units* (3.74) pertaining to one particular output time

3.72
coded displacement
coded displacement frame
coded representation of a *displacement frame* (3.73)