International Standard

ISO/IEC 23090-22

First edition
2024-09

Information technology — Coded representation of immersive media —

Part 22: Conformance for G-PCC

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

Partie 22: Conformité pour G-PCC

Reference number
ISO/IEC 23090-22:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

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

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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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. In the IEC, see www.iec.ch/understanding-standards.

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 series can be found on the ISO and IEC websites.

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

Advance in 3D capturing and rendering technologies is enabling new applications and services in the field of assisted and autonomous driving, maps, cultural heritage, industrial processes, immersive real-time communication, and Virtual/Augmented/Mixed reality (VR/AR/MR) content creation, transmission and communication. Point clouds have arisen as one of the main representations for such applications. A point cloud frame consists of a set of 3D points. Each point, in addition to having a 3D position may also be associated with numerous other attributes such as colour, transparency, reflectance, timestamp, surface normal, and classification. Such representations require a large amount of data, which can be costly in terms of storage and transmission. Therefore, ISO/IEC 23090-9 specifies Geometry-based Point Cloud Compression (G-PCC), which aims at efficiently compressing point cloud representations.

This document is the conformance testing specification for ISO/IEC 23090-9.

Information technology — Coded representation of immersive media —

Part 22: Conformance for G-PCC

1 Scope

This document specifies a set of tests and procedures designed to indicate whether encoders or decoders meet the normative requirements specified in ISO/IEC 23090-9.

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 23090-9, *Information technology — Coded representation of immersive media — Part 9: Geometry-based point cloud compression*

ISO/IEC 23090-21, *Information technology — Coded representation of immersive media — Part 21: Reference software for Geometry-based Point Cloud Compression (G-PCC)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23090-9 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
bitstream
sequence of bits

Note 1 to entry: An ISO/IEC 23090-9 G-PCC bitstream.

3.2
decoding process
process that restores information from a coded representation to the original form according to a given code

Note 1 to entry: An embodiment of the decoding process specified by ISO/IEC 23090-9.

3.3
decoder
embodiment of a *decoding process* (3.2)

Note 1 to entry: An ISO/IEC 23090-9 G-PCC decoder. The decoder does not include the rendering and display process, which is outside the scope of this document.

3.4
encoder
embodiment of a process that produces a *bitstream* (3.1)

3.5
reference software decoder
particular *decoder* (3.3) provided as a software package for use as an example available for study, as a potential starting basis for the development of other decoder, as a way of testing bitstreams for conformance to a decoding process specification, or as a reference for comparison with the behaviour of other decoder

Note 1 to entry: A software decoder is provided in ISO/IEC 23090-21.

3.6
reference software encoder
particular *encoder* (3.4) provided as a software package for use as an example available for study, as a potential starting basis for the development of other encoder, or as a reference for comparison with the behaviour of other encoder

Note 1 to entry: A software encoder is provided in ISO/IEC 23090-21.

4 Abbreviated terms

For the purposes of this document, relevant abbreviated terms are specified in ISO/IEC 23090-9:2023, Clause 4.

5 Conventions

For the purposes of this document, relevant conventions are specified in ISO/IEC 23090-9:2023, Clause 5.

6 Conformance testing for ISO/IEC 23090-9

6.1 General

The following subclauses specify the tests for verifying conformance of bitstreams as well as decoders. These tests shall use the test data (bitstream test suites) provided in https://standards.iso.org/iso-iec/23090/-22/ed-1/en and the reference software decoder is specified in Rec. ISO/IEC 23090-21.

6.2 Bitstream conformance

The bitstream conformance follows the specification text in ISO/IEC 23090-9. See Annex A.

6.3 Decoder conformance

The decoder conformance for ISO/IEC 23090-9, such as conformance point, is specified in ISO/IEC 23090-9:2023, 6.5.1.

6.4 Procedure to test bitstreams

A bitstream that claims conformance with ISO/IEC 23090-9 shall pass the following test.

The bitstream shall be decoded by processing it with the reference software decoder. When processed by the reference software decoder, the bitstream shall not cause any error or non-conformance messages to be reported by the reference software decoder. This test should not be applied to bitstreams that are known to contain errors introduced by transmission, as such errors are highly likely to result in bitstreams that lack conformance to ISO/IEC 23090-9.

Successfully passing the reference software decoder test provides only a strong presumption that the bitstream under test does indeed meet all the requirements (except Annex C) specified in ISO/IEC 23090-9 that are tested by the reference software decoder.

ISO/IEC 23090-9 contains several informative recommendations that are not an integral part of that International Standard. When testing a bitstream for conformance, it can also be useful to test whether or not the bitstream follows those recommendations.

To check correctness of a bitstream, it is necessary to parse the entire bitstream and to extract all the syntax elements and other values derived from those syntactic elements and used by the decoding process specified in ISO/IEC 23090-9.

A verifier may not necessarily perform all stages of the decoding process specified in ISO/IEC 23090-9 in order to verify bitstream correctness. Many tests can be performed on syntax elements in a state prior to their use in some processing stages.

6.5 Procedure to test decoder conformance

6.5.1 Conformance bitstreams

A bitstream has values of main_profile_compatibility_flag and level_idc corresponding to a set of specified constraints on a bitstream for which a decoder conforming to a specified profile, and level is required in ISO/IEC 23090-9:2023, Annex A to properly perform the decoding process.

6.5.2 Contents of the bitstream file

The conformance bitstreams are included in this document as an electronic attachment. The following information is included in a single folder for each such bitstream.

— *.bit – bitstream as described in subclause 6.6.2 (mandatory)

— readme.md – description (mandatory)

— *.cfg – config file used to generate bitstream with TMC13 encoder SW (optional, not applicable if TMC13 encoder release version not used)

— *.md5 – MD5sum of the bitstream file (mandatory)

— *_dec.ply – unordered decoded point cloud frames (optional).

— *_dec.ply.md5 – MD5 checksum for *_dec.ply (optional)

— Makefile – Script to regenerate the bitstream (optional)

NOTE Reference software decoder can be used to generate _*dec.ply file

6.5.3 Requirements on output of the decoding process

The output of the decoding process is specified in ISO/IEC 23090-9:2023, Clause 8.

A decoder shall be configured to output integer point position in accordance with ISO/IEC 23090-9:2023, subclause 6.5.1 so that that the decoder output data in conformance testing mode.

The rendering process, which may follow the output of the decoding process, is outside the scope of this document.

6.5.4 Recommendations

In addition to the requirements, it is desirable that conforming decoders implement various informative recommendations specified in ISO/IEC 23090-9 that are not an integral part of that International Standard. This clause discusses some of these recommendations.

It is recommended that a conforming decoder be able to resume the decoding process as soon as possible after the loss or corruption of part of a bitstream.

6.6 Specification of the test bitstreams

6.6.1 General

The bitstreams used for the decoder conformance testing specified in this document shall be those listed in Table 1. Characteristics of each bitstream are specified in this subclause.

Table 1 — List of reference bitstreams.

Categories	Feature Name	Features tested
Common functionality	EBS	Entropy bypass stream
	EC	Entropy continuation
	ST	Slice/Tile
	GPS	GPS (Geometry Parameter Set)
	APS	APS (Attribute Parameter Set)
Geometry coding	GTT	Geom tree coding type
	GSO	Geometry scaling in Occupancy tree
	OG	Occupancy tree geometry
	DPO	Duplicate points in Occupancy tree
	IDCM	IDCM
	NAV	Neighbour availability volume (N3/N6)
	QTBT	QTBT
	IOC	Intra occupancy contexts
	ACN	Adjacent child neighbours
	PLANAR	Planar
	AngIDCM	Angular-IDCM
	AngPLANAR	Angular-Planar
	PredGEO	Predictive geometry
	PredDUP	Predictive duplicate
	PredANG	Predictive geometry: Angular mode
	PredGS	Predictive geometry: Geometry Scaling
	CAPS	Change APS

Table 1 *(continued)*

Categories	Feature Name	Features tested
Attribute coding	GATT	Generic attribute
	QuantATT	Quantization
	LodRAHT	LoD and RAHT
	SLOD	Single LoD
	NumLOD	Number of LoDs
	DecLOD	LoD method: Decimation
	DisLOD	LoD method: Distance
	CentLOD	LoD method: Block-based
	SL	Scalable Lifting
	PSInterLOD	Predictor search: inter LoD
	PSIntraLOD	Predictor search: intra LoD
	NumPRED	Number of predictors
	DirectPRED	Predicting Transform: Direct predictors
	ACO	Axis coding order
	NB	Neighbour bias
	LCP	Lifting Transform: Last component pred
	ICP	Predicting Transform: Inter component pred
	RahtPRED	RAHT: prediction
	SDC	Spherical domain coding
	ANBF	Attribute neighbour blending filtering

6.6.2 Test bitstreams

6.6.2.1 Entropy bypass stream (EBS)

6.6.2.1.1 Test bitstream EBS_A_Panasonic[1)]

Specification: The bitstream exercise the entropy bypass stream On functionality.

— bypass_stream_enabled equal to 1.

— attr_coding_type is equal to 0 (RAHT).

Functional stage: Entropy bypass stream enabled and RAHT Attribute Coding on Simple profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy stream enabled.

6.6.2.1.2 Test bitstream EBS_B_Panasonic

Specification: The bitstream exercise the entropy bypass stream On functionality.

— bypass_stream_enabled equal to 1.

— attr_coding_type is equal to 1 (LoD with Predicting Transform).

Functional stage: Entropy bypass stream enabled and Predicting Attribute Coding on Predictive profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy stream enabled.

1) This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of the product named.

6.6.2.1.3 Test bitstream EBS_C_Panasonic

Specification: The bitstream exercise the entropy bypass stream Off functionality.

— bypass_stream_enabled equal to 0.

— attr_coding_type is equal to 1 (LoD with Predicting Transform).

Functional stage: Entropy bypass stream disabled and Predicting Attribute Coding on Dense profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy stream disabled.

6.6.2.1.4 Test bitstream EBS_D_Panasonic

Specification: The bitstream exercise the entropy bypass stream On functionality.

— bypass_stream_enabled equal to 1.

— attr_coding_type is equal to 1 (LoD with Predicting Transform).

Functional stage: Entropy bypass stream enabled and Predicting Attribute Coding on Dense profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy stream enabled.

6.6.2.1.5 Test bitstream EBS_E_Panasonic

Specification: The bitstream exercise the entropy bypass stream On functionality.

— bypass_stream_enabled equal to 1.

— attr_coding_type is equal to 2 (LoD with Lifting Transform).

Functional stage: Entropy bypass stream enabled and Lifting Attribute Coding on Main profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy stream enabled.

6.6.2.2 Entropy continuation (EC)

6.6.2.2.1 Test bitstream EC_A_Panasonic

Specification: The bitstream exercise the entropy continuation functionality on each coding method.

— slice_reordering_constraint is equal to 1.

— entropy_continuation_enabled is equal to 1.

— slice_entropy_continuation is used.

— geom_tree_type is equal to 0 (occupancy tree).

— occtree_bitwise_coding is equal to 1.

— attr_coding_type is equal to 0 (RAHT).

Functional stage: Entropy continuation on Occupancy tree Geometry Coding and RAHT Attribute Coding on Dense profile and Main profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is on each coding method.

6.6.2.2.2 Test bitstream EC_B_Panasonic

Specification: The bitstream exercise the entropy continuation functionality on each coding method.

— slice_reordering_constraint is equal to 1.

— entropy_continuation_enabled is equal to 1.

— slice_entropy_continuation is used.

— geom_tree_type is equal to 0 (occupancy tree).

— occtree_bitwise_coding is equal to 0 (using the dictionary encoding).

— attr_coding_type is equal to 2 (LoD with Lifting Transform).

Functional stage: Entropy continuation on Occupancy tree Geometry Coding with dictionary encoding and Lifting Attribute Coding on Simple profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is on each coding method.

6.6.2.2.3 Test bitstream EC_C_Panasonic

Specification: The bitstream exercise the entropy continuation functionality on each coding method.

— slice_reordering_constraint is equal to 1.

— entropy_continuation_enabled is equal to 1.

— slice_entropy_continuation is used.

— geom_tree_type is equal to 1 (geometry_predtree).

— attr_coding_type is equal to 1 (LoD with Predicting Transform).

Functional stage: Predictive Geometry Coding and LoD with Predicting Transform Attribute Coding on Predictive profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is on each coding method.

6.6.2.2.4 Test bitstream EC_D_Panasonic

Specification: The bitstream exercise the entropy continuation functionality on each coding method.

— slice_reordering_constraint is equal to 1.

— entropy_continuation_enabled is equal to 0, 1, 1, 0, 1.

— slice_entropy_continuation is used.

Functional stage: Predictive Geometry Coding and LoD with Predicting Transform Attribute Coding on Predictive profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is on each coding method.

6.6.2.3 Slice/Tile (ST)

6.6.2.3.1 Test bitstream ST_A_LGE

Specification: The bitstream consists of zero tile and one slice.

— Number of tiles is equal to 0.

— Number of slices is equal to 1.

Functional stage: Common functionality in Simple profile.

Purpose: Check that the decoder can properly decode bitstreams in which zero tile and one slice has.

6.6.2.3.2 Test bitstream ST_B_LGE

Specification: The bitstream consists of zero tile and one slice.

— Number of tiles is equal to 0.

— Number of slices is equal to 1.

Functional stage: Common functionality in Predictive profile.

Purpose: Check that the decoder can properly decode bitstreams in which zero tile and one slice has.

6.6.2.3.3 Test bitstream ST_C_LGE

Specification: The bitstream consists of zero tile and one slice.

— Number of tiles is equal to 0.

— Number of slices is equal to 1.

Functional stage: Common functionality in Main profile and Dense profile.

Purpose: Check that the decoder can properly decode bitstreams in which zero tile and one slice.

6.6.2.3.4 Test bitstream ST_D_LGE

Specification: The bitstream consists of zero tile and multiple slices.

— Number of tiles is equal to 0.

— Number of slices is equal to 4.

Functional stage: Common functionality in Simple profile.

Purpose: Check that the decoder can properly decode bitstreams in which zero tile and multiple slices have.

6.6.2.3.5 Test bitstream ST_E_LGE

Specification: The bitstream consists of zero tile and multiple slices.

— Number of tiles is equal to 0.

— Number of slices is equal to 4.

Functional stage: Common functionality in Predictive profile.

Purpose: Check that the decoder can properly decode bitstreams in which zero tile and multiple slices have.