International Standard

ISO/IEC 23090-35

First edition
2026-11

Information technology — Coded representation of immersive media —

Part 35: Conformance and reference software for low latency, low complexity LiDAR coding

Sample Document
get full document from standards.iteh.ai

Reference number
ISO/IEC 23090-35:2026(en)

© ISO/IEC 2026

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

Sample Document

get full document from standards.iteh.ai



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2026

All rights reserved.

This ISO publication is protected by copyright and is owned by ISO and/or its licensors.

The content of this ISO publication is provided under licence, not sold. Use is subject to the applicable licence terms issued by ISO, an ISO member body, or an authorized third-party distributor.

Except as required for implementation or expressly permitted by a separate licence, no part of this ISO publication may be reproduced, distributed, modified, used, or made available in any form or by any means – electronic or mechanical, including photocopying, scanning, recording, or posting on internal or external digital platforms.

Any use beyond the scope of the granted rights is prohibited and may result in legal action.

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

Published in Switzerland

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

Contents

Page

Foreword ..... **iv**
Introduction ..... **v**
1 Scope **1**
2 Normative references **1**
3 Terms and definitions **1**
4 Abbreviated terms **2**
5 Conventions **2**
6 Reference software for ISO/IEC 23090-30 ..... **2**
7 Conformance testing for ISO/IEC 23090-30 ..... **2**
7.1 General 2
7.2 Bitstream conformance 2
7.3 Decoder conformance 2
7.4 Procedure to test bitstreams 2
7.5 Procedure to test decoder conformance 3
7.5.1 Conformance bitstreams 3
7.5.2 Contents of the bitstream file 3
7.5.3 Requirements on output of the decoding process 3
7.5.4 Recommendations 3
7.6 Specification of the test bitstreams 4
7.6.1 General 4
7.6.2 Test bitstreams 5

Sample Document
get full document from standards.iteh.ai

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

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.

Sample Document

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*.

get full document from standards.iteh.ai

A list of all parts in the ISO/IEC 23090 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.

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

Introduction

0.1 General

This document specifies reference software and conformance testing for ISO/IEC 23090-30 low latency, low complexity LiDAR coding (L3C2).

0.2 Purpose

The purpose of this document is to provide the following:

— Reference decoder software for decoding ISO/IEC 23090-30 compliant bitstreams.

— Reference encoder software for generating ISO/IEC 23090-30 compliant bitstreams.

— Procedures to test bitstream and decoder conformance.

— A set of reference bitstreams conforming to ISO/IEC 23090-30.

0.3 Examples of use

Some examples of uses that can be appropriate for the reference decoder software are as follows:

— As an illustration of how to perform the decoding process specified in ISO/IEC 23090-30.

— As the starting basis for the implementation of a decoder that conforms to ISO/IEC 23090-30.

— For testing the conformance of a decoder implementation with the decoding process specified in ISO/IEC 23090-30.

— For testing the conformance of a bitstream to the constraints specified for bitstream conformance in ISO/IEC 23090-30, as the software can detect and report many bitstream conformance violations.

NOTE 1 The lack of the detection of any conformance violation by the reference decoder software is not definitive proof that the bitstream conforms to all constraints specified for bitstream conformance in ISO/IEC 23090-30.

Some examples of uses that can be appropriate for the reference encoder software are as follows:

— As an illustration of how to perform an encoding process that produces bitstreams that conform to the constraints specified for bitstream conformance in ISO/IEC 23090-30.

— As the starting basis for the implementation of an encoder that conforms to ISO/IEC 23090-30.

— As a means of generating bitstreams for testing the conformance of a decoder implementation with the decoding process specified in ISO/IEC 23090-30.

— As a means of evaluating and demonstrating examples of the quality that can be achieved by an encoding process that conforms to ISO/IEC 23090-30.

NOTE 2 No guarantee of the quality that will be achieved by an encoder is provided by its conformance to ISO/IEC 23090-30, as the conformance of an encoder to ISO/IEC 23090-30 is defined only in terms of format constraints imposed on the bitstream syntax. The reference encoder software provides some illustrative examples of what quality can be achieved in conformance to ISO/IEC 23090-30, it does not define minimum encoding quality or maximum encoding quality.

Sample Document
Get full document from standards.iteh.ai

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

Sample Document

get full document from standards.iteh.ai

Information technology — Coded representation of immersive media —

Part 35: Conformance and reference software for low latency, low complexity LiDAR coding

1 Scope

This document provides accompanying reference software for ISO/IEC 23090-30. The software is an integral part of this document.

The use of this reference software is not required to make an implementation of an encoder or decoder in conformance to ISO/IEC 23090-30. Requirements established in ISO/IEC 23090-30 take precedence over the behaviour of the reference software.

This document also specifies a set of tests and procedures designed to indicate whether encoders or decoders meet the normative requirements specified in ISO/IEC 23090-30.

Sample Document

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.

iTeh Standards (https://standards.iteh.ai) Document Preview

ISO/IEC 23090-30:2026, *Information technology — Coded representation of immersive media — Part 30: Low latency, low complexity light detection and ranging (LiDAR) coding*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23090-30 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 representing compressed point cloud data structured according to the syntax of a specific coding specification

3.2
decoder
process or device that reconstructs point cloud data from a structured bitstream conforming to a specific coding specification

Note 1 to entry: Decoder does not include the point cloud rendering process, which is outside the scope of this document.

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

3.3
encoder
embodiment of a process that produces a bitstream

3.4
reference software decoder
software implementation of the decoding process that reconstructs point cloud data from a structured bitstream conforming to a specific coding specification

3.5
reference software encoder
software implementation that compresses point cloud data into a structured bitstream conforming to a specific coding specification

4 Abbreviated terms

The abbreviated terms in ISO/IEC 23090-30:2026, Clause 4 apply.

5 Conventions

The conventions specified in ISO/IEC 23090-30:2026, Clause 5 apply.

6 Reference software for ISO/IEC 23090-30

Reference software is useful in aiding users of a compression standard to establish and test conformance and interoperability, and to educate users and demonstrate the capabilities of ISO/IEC 23090-30:2026. The reference software for ISO/IEC 23090-30:2026 is available at: https://standards.iso.org/iso-iec/23090/-35/ed-1/en/.

Sample Document
get full document from standards.iteh.ai

The software package contains one part:

— mpeg-pcc-tml software: Support for Simple and Main profiles.

The reference software manual is mpeg-pcc-tml-sw-manual.pdf. This manual can be used to install and use the reference software.

7 Conformance testing for ISO/IEC 23090-30

7.1 General

The following subclauses specify normative tests for verifying conformance of bitstreams as well as decoders. Those normative tests make use of test data (bitstream test suites) provided are available at: https://standards.iso.org/iso-iec/23090/-35/ed-1/en/ and the reference software decoder specified in Clause 6.

7.2 Bitstream conformance

Bitstream conformance follows the specification text in ISO/IEC 23090-30.

7.3 Decoder conformance

Decoder conformance for ISO/IEC 23090-30, such as conformance point, etc., is specified by ISO/IEC 23090-30:2026, 6.5.1.

7.4 Procedure to test bitstreams

A bitstream that claims conformance with ISO/IEC 23090-30 shall pass the following normative test.

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

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 23090-30.

Successfully passing the reference software decoder test provides only a strong presumption that the bitstream under test does indeed meet all the requirements specified in ISO/IEC 23090-30 that are tested by the reference software decoder.

ISO/IEC 23090-30 contains several informative recommendations that are not an integral part of this document. 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 23090-30.

A verifier need not necessarily perform all stages of the decoding process specified in ISO/IEC 23090-30 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.

7.5 Procedure to test decoder conformance

7.5.1 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 Annex A of ISO/IEC 23090-30:2026 to properly perform the decoding process.

Sample Document

7.5.2 Contents of the bitstream file

get full document from standards.iteh.ai

The conformance bitstreams are included in this document and are available at: https://standards.iso.org/iso-iec/23090/-35/ed-1/en/. The following information is included in a single zipped file for each such bitstream.

— *.bit – bitstream as described in subclause 7.6.2

— readme.md – description

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

— *.md5 – MD5sum of the bitstream file

— *_dec.ply – unordered decoded point cloud frames

— *_dec.ply.md5 – MD5 checksum for decoded point cloud frame

7.5.3 Requirements on output of the decoding process

The output of the decoding process is specified in Clause 8 of ISO/IEC 23090-30:2026.

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

7.5.4 Recommendations

In addition to the requirements, it is desirable that conforming decoders implement various informative recommendations specified in ISO/IEC 23090-30 that are not an integral part of ISO/IEC 23090-30. This clause discusses some of these recommendations.

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

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.

7.6 Specification of the test bitstreams

7.6.1 General

The test bitstreams are part of the mechanism to verify decoder conformance according to the ISO/IEC 23090-30 specification. The bitstreams used for the decoder conformance testing specified in this document shall be those listed in Table 1. They are available at: https://standards.iso.org/iso-iec/23090/-35/ed-1/en/.

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)
	ACO	Axis coding order
Geometry coding	GS	Geometry scaling
	OGC	Onechain geometry coding
	PLCS	Preserve laser coding state
	DPOC	Duplicate points in onechain geometry coding
	UNEOC	Use NSensing estimation in onechain coding
	UCN	Use Coarse neighbours
	UVP	Use Vertical prediction
	LLA	Low latency attributes coding
	UPOC	Unordered points in onechain coding
	SCS	Sensing coverage signalling
Attribute coding	GATT	Generic attribute
	QuantATT	Quantization
	LodRAHT	Level of Details (LoD) and Region Adaptive Hierarchical Transform (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	Direct predictors (Pred)
	NB	Neighbour bias
	LCP	Last component pred
	ICP	Predicting Transform: Inter component pred

Sample Document
get full document from standards.iteh.ai

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

Table 1 *(continued)*

Categories	Feature Name	Features tested
	RahtPRED	RAHT prediction
	SDC	Spherical domain coding
	ANBF	Attribute neighbour blending filtering

7.6.2 Test bitstreams

7.6.2.1 Entropy bypass stream (EBS)

7.6.2.1.1 Test bitstream EBS_A_X

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

— bypass_stream_enabled flag equal to 0.

— lowLatencyAttributesCoding equal to 1 (low latency attribute).

— attr_coding_type is equal to 4 (onechain).

Functional stage: Entropy bypass stream disabled and low latency attribute coding on Main profile.

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

7.6.2.1.2 Test bitstream EBS_B_X

Sample Document

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

— bypass_stream_enabled flag equal to 1

— lowLatencyAttributesCoding equal to 1 (low latency attribute)

get full document from standards.iteh.ai

— attr_coding_type is equal to 4 (onechain)

Functional stage: Entropy bypass stream enabled and low latency attribute coding on Main profile.

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

7.6.2.1.3 Test bitstream EBS_C_X

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

— bypass_stream_enabled flag equal to 1

— lowLatencyAttributesCoding equal to 0

— attr_coding_type is equal to 0 (RAHT)

Functional stage: Entropy bypass stream enabled and RAHT attribute coding on Simple and Main profile.

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

7.6.2.1.4 Test bitstream EBS_D_X

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

— bypass_stream_enabled flag equal to 1

— lowLatencyAttributesCoding equal to 0

— attr_coding_type is equal to 1 (Predicting)

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

Functional stage: Entropy bypass stream enabled and Predicting attribute coding on Simple and Main profile.

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

7.6.2.1.5 Test bitstream EBS_E_X

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

— bypass_stream_enabled flag equal to 1

— lowLatencyAttributesCoding equal to 0

— attr_coding_type is equal to 2 (Lifting)

Functional stage: Entropy bypass stream enabled and Lifting attribute coding on Simple and Main profile.

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

7.6.2.1.6 Test bitstream EBS_F_X

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

— bypass_stream_enabled flag equal to 1

— lowLatencyAttributesCoding equal to 0

— attr_coding_type is equal to 3 (raw)

Functional stage: Entropy bypass stream enabled and raw attribute coding on Simple and Main profile.

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

Sample Document
get full document from standards.iteh.ai

7.6.2.1.7 Test bitstream EBS_G_X

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

— bypass_stream_enabled flag equal to 0

— lowLatencyAttributesCoding equal to 0

— attr_coding_type is equal to 0 (RAHT)

Functional stage: Entropy bypass stream disabled and RAHT attribute coding on Main profile.

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

7.6.2.1.8 Test bitstream EBS_H_X

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

— bypass_stream_enabled flag equal to 0

— lowLatencyAttributesCoding equal to 0

— attr_coding_type is equal to 1 (Predicting)

Functional stage: Entropy bypass stream disabled and Predicting attribute coding on Main profile.

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

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

7.6.2.1.9 Test bitstream EBS_I_X

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

— bypass_stream_enabled flag equal to 0

— lowLatencyAttributesCoding equal to 0

— attr_coding_type is equal to 2 (Lifting)

Functional stage: Entropy bypass stream disabled and Lifting attribute coding on Main profile.

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

7.6.2.1.10 Test bitstream EBS_J_X

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

— bypass_stream_enabled flag equal to 0

— lowLatencyAttributesCoding equal to 0

— attr_coding_type is equal to 3 (raw)

Functional stage: Entropy bypass stream disabled and raw attribute coding on Main profile.

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

7.6.2.2 Entropy continuation (EC)

Sample Document

7.6.2.2.1 Test bitstream EC_A_X

get full document from standards.iteh.ai

Specification: The bitstream exercises the entropy continuation functionality enabled.

— gbh_entropy_continuation_flag: 1

— lowLatencyAttributesCoding equal to 1 (low latency attribute)

— attr_coding_type: 4 (onechain)

— sliceMaxPoints: 25000

— sliceMinPoints: 5000

Functional stage: Entropy continuation on Onechain Geometry Coding and low latency attribute coding on Main profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is enabled.

7.6.2.2.2 Test bitstream EC_B_X

Specification: The bitstream exercises the entropy continuation functionality enabled.

— gbh_entropy_continuation_flag: 1

— lowLatencyAttributesCoding equal to 0

— attr_coding_type: 1 (predicting)

— sliceMaxPoints: 25000

— sliceMinPoints: 5000

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

Functional stage: Entropy continuation on Onechain Geometry Coding and predicting attribute coding on Simple and Main profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is enabled.

7.6.2.2.3 Test bitstream EC_C_X

Specification: The bitstream exercises the entropy continuation functionality enabled.

— gbh_entropy_continuation_flag: 1

— lowLatencyAttributesCoding equal to 0

— attr_coding_type: 0 (RAHT)

— sliceMaxPoints: 25000

— sliceMinPoints: 5000

Functional stage: Entropy continuation on Onechain Geometry Coding and RAHT attribute coding on Simple and Main profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is enabled.

7.6.2.2.4 Test bitstream EC_D_X

Specification: The bitstream exercises the entropy continuation functionality enabled.

Sample Document

— gbh_entropy_continuation_flag: 1

— lowLatencyAttributesCoding equal to 0

get full document from standards.iteh.ai

— attr_coding_type: 2 (lifting)

— sliceMaxPoints: 25000

— sliceMinPoints: 5000

Functional stage: Entropy continuation on Onechain Geometry Coding and lifting attribute coding on Simple and Main profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is enabled.

7.6.2.2.5 Test bitstream EC_E_X

Specification: The bitstream exercises the entropy continuation functionality enabled.

— gbh_entropy_continuation_flag: 1

— lowLatencyAttributesCoding equal to 0

— attr_coding_type: 3 (raw)

— sliceMaxPoints: 25000

— sliceMinPoints: 5000

Functional stage: Entropy continuation on Onechain Geometry Coding and raw attribute coding on Simple and Main profile.

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is enabled.

7.6.2.2.6 Test bitstream EC_F_X

Specification: The bitstream exercises the entropy continuation functionality enabled/disabled on different slices.

— gbh_entropy_continuation_flag: 0, 1, 0, 1

— lowLatencyAttributesCoding equal to 1 (low latency attribute)

— attr_coding_type: 4 (onechain)

— sliceMaxPoints: 25000

— sliceMinPoints: 5000

Functional stage: Entropy continuation on Onechain Geometry Coding and low latency attribute coding on Main profile.

Purpose: Check that the decoder can properly decode bitstreams in which entropy continuation function is enabled/disabled on different slices.

7.6.2.3 Slice/Tile (ST)

7.6.2.3.1 Test bitstream ST_A_X

Sample Document

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

— tileSize: 0

— partitionMethod: 0

get full document from standards.iteh.ai

Functional stage: Common functionality in Simple profile.

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

7.6.2.3.2 Test bitstream ST_B_X

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

— tileSize:0

— partitionMethod: 0

Functional stage: Common functionality in Main profile.

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

7.6.2.3.3 Test bitstream ST_C_X

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

— tileSize:0

— partitionMethod: 4

Functional stage: Common functionality in Simple profile.

© ISO/IEC 2026 – All rights reserved

ISO/IEC 23090-35:2026 • https://standards.iteh.ai/catalog/standards/iso/28143b58-fe1e-4b51-a83b-c9d6bde6d6ff/iso-iec-23090-35-2026