



**International
Standard**

ISO/IEC 23090-30

**Information technology — Coded
representation of immersive
media —**

**Part 30:
Low latency, low complexity light
detection and ranging (LiDAR)
coding**

**First edition
2026-07**

Sample Document

get full document from standards.iteh.ai



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2026

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Foreword.....	viii
Introduction	ix
1 Scope	1
2 Normative references	1
3 Terms and definitions.....	1
3.1 General terms	1
3.2 High-level syntax and entropy coding terms	4
3.3 Tree structure terms.....	7
3.4 Geometry coding terms.....	7
3.5 Attribute coding terms.....	7
4 Abbreviated terms and symbols	8
4.1 Abbreviated terms	8
4.2 Symbols.....	8
4.3 Mnemonics	9
5 Conventions	9
5.1 General.....	9
5.2 Numerical representation	9
5.3 Arithmetic operators	9
5.4 Logical operators	10
5.5 Relational operators	10
5.6 Bit-wise operators	10
5.7 Assignment operators	11
5.8 Range notation	11
5.9 Mathematical functions	11
5.9.1 General	11
5.9.2 IntCos and IntSin.....	12
5.9.3 IntSqrt.....	12
5.9.4 IntRecipSqrt	13
5.9.5 Div	14
5.9.6 Morton	14
5.9.7 FromMorton	14
5.10 Order of operation precedence	15
5.11 Named expressions.....	15
5.11.1 General.....	15
5.11.2 Scope.....	16
5.11.3 Arguments of named expressions.....	17
5.11.4 Sub-expressions	17
5.11.5 Definitions with multiple statements.....	17
5.11.6 Textual definitions	17
5.12 Variables, syntax elements and tables.....	17
6 Point cloud format and relationship to coded and output representations	18
6.1 General format	18
6.2 Attributes	18
6.2.1 General	18

6.2.2 Colour	18
6.2.3 Opacity.....	19
6.2.4 Reflectance.....	19
6.2.5 Normal vector	19
6.2.6 Material identifier	19
6.2.7 Frame number/index.....	19
6.2.8 User defined attributes	20
6.3 Codec-derived attributes.....	20
6.3.1 General	20
6.3.2 Slice identifier.....	20
6.3.3 Slice tag	20
6.3.4 Canonical point order	20
6.3.5 Point Morton order	21
6.4 Coded point cloud format.....	21
6.4.1 Sequence coordinate system	21
6.4.2 Coding coordinate system	22
6.4.3 Coded point cloud sequence.....	22
6.4.4 Coded point cloud frame.....	23
6.4.5 Slice of a coded point cloud frame.....	23
6.4.6 Repetition of slices.....	24
6.4.7 Relationship between tiles and slices	24
6.4.8 Parameter sets.....	25
6.5 Output point cloud format.....	25
6.5.1 General	25
6.5.2 Coordinate system	25
6.5.3 Fixed-point conformance output	25
6.5.4 Attributes	26
6.5.5 Output point cloud sequence	26
6.5.6 Output point cloud frame	26
7 Syntax and semantics	26
7.1 Method of specifying syntax in tabular form	26
7.2 Specification of syntax functions and descriptors	27
7.3 Syntax in tabular form.....	28
7.3.1 General	28
7.3.2 Parameter sets, ancillary data and byte alignment	28
7.3.3 Geometry data unit	35
7.3.4 Attribute data unit	38
7.3.5 Defaulted attribute data unit syntax	40
7.4 Semantics	41
7.4.1 General	41
7.4.2 Parameter sets, ancillary data and byte alignment	41
7.4.3 Geometry data unit	53
7.4.4 Attribute data unit	55
7.4.5 Defaulted attribute data unit semantics	55
8 Decoding process.....	56
8.1 General decoding process	56
8.2 Frame decoding processes	56
8.2.1 General	56
8.2.2 Frame counter	56
8.3 Slice decoding processes	56

8.3.1 General	56
8.3.2 State variables	56
8.3.3 Geometry decoding process	57
8.3.4 Default attribute values	57
8.3.5 Attribute decoding process.....	57
8.3.6 At the end of a slice	57
9 Slice geometry.....	57
9.1 General.....	57
9.2 Sensed point sequence.....	58
9.2.1 General.....	58
9.2.2 State variables and expressions.....	58
9.2.3 Syntax element semantics.....	59
9.2.4 Contextualization of decoding elevation_offset.....	62
9.2.5 Reconstruction of sensed point positions	65
9.2.6 Reconstructed output point positions	74
9.2.7 Reconstructed output low latency attributes values.....	74
9.2.8 Reconstructed point positions for attributes decoding	74
10 Slice attributes	74
10.1 General	74
10.2 Point coordinates	74
10.2.1 General.....	74
10.2.2 Conversion to scaled angular coordinates.....	75
10.3 Syntax element semantics	75
10.4 Raw attribute decoding.....	76
10.5 Attribute decoding using the region-adaptive hierarchical transform	76
10.5.1 General.....	76
10.5.2 Transform tree.....	76
10.5.3 Coefficient order	79
10.5.4 Coefficient scaling.....	80
10.5.5 Transform domain prediction	82
10.5.6 Inverse transform.....	86
10.5.7 Reconstructed attribute values	88
10.6 Attribute decoding using levels of detail	88
10.6.1 General.....	88
10.6.2 Syntax element semantics	89
10.6.3 Reconstruction process	89
10.6.4 State variables.....	89
10.6.5 Levels of detail	90
10.6.6 Predictor search.....	96
10.6.7 Reconstruction of attribute values.....	102
10.6.8 Prediction mode coding.....	103
10.6.9 Scaling	105
10.6.10 Coefficient prediction	105
10.6.11 Transform coefficient weights.....	106
10.6.12 Transform	107
10.7 Attribute quantization parameters	108
10.7.1 Syntax element semantics	108
10.7.2 Per-point regional QP offset.....	109
10.7.3 Attribute coefficient QP	109
10.7.4 Definition of <i>AttrQstep</i>	109

11 Parsing process.....	110
11.1 General	110
11.2 Data unit buffer	111
11.2.1 General.....	111
11.2.2 State	111
11.2.3 Initialization at the start of parsing a data unit.....	112
11.2.4 Initialization at the start of parsing a geometry data unit footer.....	112
11.2.5 Definition of <i>DuNextBit</i>	112
11.3 Chunked bytestream parsing.....	112
11.3.1 General.....	112
11.3.2 Chunk syntax	113
11.3.3 Chunk semantics	113
11.3.4 State	113
11.3.5 Span of chunked bytestream data within a data unit.....	113
11.3.6 The chunk buffer	114
11.3.7 State update at the start of every CBS.....	114
11.3.8 Unpacking a single chunk	114
11.3.9 Definition of <i>ChunkNextAeBit</i>	114
11.3.10 Definition of <i>ChunkNextBpBit</i>	115
11.3.11 Boundary between spliced chunked bytestreams	115
11.3.12 Location of chunked bytestream boundaries	116
11.4 General inverse binarization processes.....	116
11.4.1 Parsing unsigned fixed-length codes (FL).....	116
11.4.2 Parsing signed fixed-length codes (FL+S)	116
11.4.3 Parsing <i>k</i> -th order exp-Golomb codes (EGk)	117
11.4.4 Parsing concatenated truncated unary and <i>k</i> -th order exp-Golomb codes (TU+EGk)	117
11.4.5 Parsing concatenated bounded truncated unary and <i>k</i> -th order exp-Golomb codes (BTU+EGk).....	117
11.4.6 Parsing signed concatenated truncated unary and <i>k</i> -th order exp-Golomb codes (TU+EGk+S).....	118
11.4.7 Parsing signed concatenated bounded truncated unary and <i>k</i> -th order exp-Golomb codes (BTU+EGk+S)	118
11.4.8 Parsing signed concatenated asymmetrically bounded truncated unary and <i>k</i> -th order exp-Golomb codes (ABTU+EGk+S)	119
11.4.9 Parsing truncated unary codes (TU).....	120
11.4.10 Mapping process for signed codes	120
11.4.11 Parsing ASN.1 object identifiers	120
11.5 CABAC parsing processes	121
11.5.1 Initialization	121
11.5.2 Definition of <i>AeReadBin</i>	121
11.5.3 Contextual probability models	121
11.5.4 Arithmetic decoding engine	125
11.6 Sensor Coding State	127
11.6.1 State variable.....	127
11.6.2 Initialization at the start of a GDU	129
11.6.3 Initialization with decoded coarse position of first sensed point.....	130
11.6.4 Update with azimuth_offset.....	131
11.6.5 Update while decoding elevation_offset.....	132
11.6.6 Update with sensed point coarse position.....	132
11.7 Parsing state memorization and restoration	133

11.7.1 General.....	133
11.7.2 Geometry data units.....	133
11.7.3 Attribute data units.....	134
11.7.4 Defaulted attribute data units.....	134
12 OBUF parsing process.....	134
12.1 General	134
12.2 Creation of an OBUF instance.....	134
12.2.1 Creation and initialization of OBUF trees	135
12.2.2 Creation and initialization of an array of OBUF ACPMs	136
12.3 Call and update of an OBUF instance	136
12.3.1 Decode and update according to OBUF trees.....	136
12.4 Decode of a bin based on an OBUF ACPM.....	137
Annex A (normative) Profiles and levels.....	139
Annex B (normative) Type-length-value encapsulated bytestream format.....	144
Annex C (informative) Arithmetic encoding engine	146
Annex D (informative) Index of named expressions and variables.....	149

Sample Document

get full document from standards.iteh.ai

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 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, 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 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

Advancements in 3D capturing and rendering technologies are enabling new applications and services in the fields of assisted and autonomous driving, cartography, and other industrial processes. Point clouds captured with LiDAR (Light Detection And Ranging) sensors, sometimes coupled with traditional camera sensors, have arisen as one of the main representations for such applications. A point cloud frame consists of a set of 3D points. Every point, in addition to having a 3D position, may also be associated with numerous other attributes such as intensity, colour, timestamp, and classification. Such representations require a large amount of data, which can be costly in terms of storage and transmission and are often used in real-time application which requires fast processing of the point cloud data. This document provides the method for efficiently compressing point cloud representations while ensuring low complexity and low latency constraints can be met.

Sample Document

get full document from standards.iteh.ai

Sample Document

get full document from standards.iteh.ai

Information technology — Coded representation of immersive media —

Part 30:

Low latency, low complexity light detection and ranging (LiDAR) coding

1 Scope

This document specifies low latency, low complexity light detection and ranging (LiDAR) coding.

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 8825-1 | Rec. ITU-T X.690, *Information technology — ASN.1 encoding rules — Part 1: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

ISO/IEC 9834-1 | Rec. ITU-T X.660, *Information technology — Procedures for the operation of object identifier registration authorities — Part 1: General procedures and top arcs of the international object identifier tree*

ISO/IEC 9834-8 | Rec. ITU-T X.667, *Information technology — Procedures for the operation of object identifier registration authorities — Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers*

ISO/IEC 23091-2, *Information technology — Coding-independent code points — Part 2: Video*

Recommendation ITU-T T.35, *Procedure for the allocation of ITU-T defined codes for non-standard facilities*

3 Terms and definitions

For the purposes of this document, the following terms and definitions 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 General terms

3.1.1

point

fundamental element of a *point cloud* (3.1.2) comprising a position specified as *Cartesian coordinates* (3.1.8) and zero or more *attributes* (3.1.19)

3.1.2

point cloud

unordered list of *points* (3.1.1)

3.1.3

point cloud sequence

sequence of one or more *point clouds* (3.1.2)

3.1.4

point cloud frame

point cloud (3.1.2) in a *point cloud sequence* (3.1.3)

3.1.5

coded point cloud frame

coded representation of a *point cloud frame* (3.1.4)

3.1.6

canonical point order

canonical decoding order

order of *points* (3.1.1) decoded from a *slice* (3.1.21) according to the decoding and parsing processes specified in this document

3.1.7

bounding box

axis-aligned cuboid defining a spatial region that bounds a set of *points* (3.1.1)

3.1.8

coordinates

<Cartesian> three scalar multiples of respective orthogonal XYZ (3.1.11) unit vectors with finite precision and bounds that specify a position relative to a fixed reference

3.1.9

coordinates

<angular> a position specified as the radial distance ρ from the V axis, an azimuth angle φ in the S-T plane and an indexed elevation

3.1.10

coordinates

<attribute> either *STV* (3.1.12) or scaled *RPI* (3.1.13) point coordinates used to code an attribute

3.1.11

XYZ (axes)

X, Y and Z axes, in that order, used to represent *Cartesian coordinates* (3.1.8)

3.1.12

STV (axes)

S, T and V axes, in that order, that are a sequence-dependent permutation of the XYZ axes (3.1.11); used to represent the coded *geometry* (3.1.18)

3.1.13

RPI (axes)

R, P and I axes, in that order, used to represent *angular coordinates* (3.1.9)

3.1.14

sequence coordinate system

scaled and translated application-specific coordinate system that applies to an entire coded *point cloud sequence* (3.1.3), and in which all *points* (3.1.1) have non-negative, fixed-point coordinates

3.1.15**coding coordinate system**

scaled *sequence coordinate system* (3.1.14) that applies for an entire coded *point cloud sequence* (3.1.3), and in which all *points* (3.1.1) have non-negative integer coordinates

3.1.16**slice coordinate system**

translated *coding coordinate system* (3.1.15) that applies for a single *slice* (3.1.21), and in which all *points* (3.1.1) in the *slice* have non-negative integer coordinates

3.1.17**sensing beam**

sampler of point positions using angular coordinates by rays cast with a fixed elevation and from a point on and rotating around the V axis at the angular origin

3.1.18**geometry**

point positions (3.4.1) associated with a set of *points* (3.1.1)

3.1.19**attribute**

scalar or vector property associated with each *point* (3.1.1) in a *point cloud* (3.1.2)

EXAMPLE Colour, reflectance, frame index, etc.

3.1.20**position**

<bit> bit in a binary string or value, representing the factor 2^{position}

EXAMPLE The LSB has bit position 0.

3.1.21**slice**

geometry and attributes for part of, or an entire, *coded point cloud frame* (3.1.5)

Note 1 to entry: the bounding boxes of any two slices can intersect.

3.1.22**tile**

set of *slices* (3.1.21) identified by a common *slice_tag syntax element* value (3.2.15) whose *geometry* (3.1.18) should be contained within a *bounding box* (3.1.7) specified in a tile inventory data unit

3.1.23**Morton code**

non-negative integer obtained by interleaving the bits of three integers

3.1.24**Morton order**

elements ordered according to their *Morton code* (3.1.23)

3.1.25**sparse array**

array with fewer set elements than total addressable elements; unset elements can have an inferred value when accessed

3.1.26**coarse position**

a coarse representation in 2D plane of position of each *point* (3.1.1) in a *point cloud* (3.1.2), used to represent the coded *geometry* (3.1.18)

3.1.27

lexicographic order

dictionary order used to order a list of *points* (3.1.1) by firstly in the azimuthal angle and secondly in the elevation angle of these points, and these points are decoded in the order

3.1.28

late points

points whose order is late according to the lexicographic order due to the noise of data, or transmission delay

3.1.29

neighbour

point whose coarse position is preceding coarse position of current point according to the lexicographic order

3.1.30

sensing head

sensor unit within a LiDAR capture system that covers a limited azimuthal range, with multiple sensing heads used to cover a larger field of view

3.1.31

sensing order index

order index, per head, in which points are captured by the sensing head

3.2 High-level syntax and entropy coding terms

3.2.1

ASN.1

abstract syntax notation one

notation specified by Rec. ITU-T X.660 | ISO/IEC 9834-1 that is used for the definition of data types, values and constraints on data types

[SOURCE: Rec. ITU-T X.660 | ISO/IEC 9834-1]

3.2.2

bin

binary symbol (bit) of the *binarized* (3.2.3) representation of a *syntax element* value (3.2.15)

3.2.3

binarization

specification of a *syntax element*'s value (3.2.15) as a sequence of *bins* (3.2.2)

3.2.4

bypass

<symbol> static, equiprobable probability model

3.2.5

bypass

<stream> *bypass symbols* (3.2.4) that are not encoded in an arithmetic-coded *bitstream* (3.2.6)

3.2.6

bitstream

<data> sequence of bits

3.2.7

bitstream

<coded sequence> sequence of bits, in the form of encapsulated *data units* (3.2.12), that represents a coded *point cloud sequence* (3.1.3)

3.2.8

set bit

bit with the value 1

3.2.9

unset bit

bit with the value 0

3.2.10

byte

sequence of 8 bits, typeset with the most significant bit on the left and the least significant bit on the right

Note 1 to entry: When represented in a bitstream, the most significant bit of a byte is first.

3.2.11

byte aligned

bitstream (3.2.6) position that is an integer multiple of eight bits from the position of the first bit in the bitstream

3.2.12

data unit

DU

sequence of *bytes* (3.2.10) conveying a single *syntax structure* (3.2.16) of known length

3.2.13

data unit header

parameters, located from the start of a *data unit* (3.2.12)

3.2.14

data unit footer

parameters, located from the end of a *data unit* (3.2.12)

3.2.15

syntax element

element of data represented in the *bitstream* (3.2.6)

3.2.16

syntax structure

zero or more *syntax elements* (3.2.15) present together in the *bitstream* (3.2.6) in a specified order

3.2.17

parameter set

collection of parameters that apply when activated

3.2.18

sequence parameter set

SPS

parameters for an entire coded *point cloud sequence* (3.1.3), conveyed by an SPS *data unit* (3.2.12) and activated when referenced by a geometry data unit