

# ISO/IEC JTC 1/SC 29/WG 7 N01085

---

## ISO/IEC JTC 1/SC 29/WG 7 MPEG 3D Graphics and Haptics coding Convenorship: AFNOR (France)

---

<b>Document type:</b>	Approved WG 7 document
<b>Title:</b>	Enhanced G-PCC Test Model v29
<b>Status:</b>	Draft
<b>Date of document:</b>	2025-2-21
<b>Source:</b>	ISO/IEC JTC 1/SC 29/WG7
<b>Expected action:</b>	None
<b>Action due date:</b>	None
<b>No. of pages:</b>	25
<b>Email of convenor:</b>	<a href="mailto:marius.preda@imt.fr">marius.preda@imt.fr</a>
<b>Committee URL:</b>	<a href="https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg7">https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg7</a>

**INTERNATIONAL ORGANISATION FOR STANDARDISATION  
ORGANISATION INTERNATIONALE DE NORMALISATION  
ISO/IEC JTC 1/SC 29/WG 7  
MPEG 3D GRAPHICS AND HAPTICS CODING**

**ISO/IEC JTC 1/SC 29/WG 7 N01085  
Geneva – Jan. 2025**

*Title:* Enhanced G-PCC Test Model v29  
*Source:* ISO/IEC JTC 1/SC 29/WG7  
*Serial Number:* 24805

---

## Abstract

This document is a user manual describing usage of reference software for the G-PCC project. It applies to version 29.0 of the software.

## Contents

<b>1</b>	<b>General Information</b>	<b>6</b>
1.1	Bug reporting . . . . .	6
<b>2</b>	<b>Obtaining the software</b>	<b>6</b>
2.1	Building . . . . .	7
2.1.1	Linux . . . . .	7
2.1.2	OSX . . . . .	7
2.1.3	Windows . . . . .	7
<b>3</b>	<b>Using the codec</b>	<b>7</b>
3.1	General options . . . . .	8
3.1.1	--help . . . . .	8
3.1.2	--config=FILE, -c . . . . .	8
3.1.3	--mode=VALUE . . . . .	8
3.2	I/O parameters . . . . .	8
3.2.1	--firstFrameNum=INT-VALUE . . . . .	8
3.2.2	--frameCount=INT-VALUE . . . . .	8
3.2.3	--uncompressedDataPath=FILE . . . . .	8
3.2.4	--compressedStreamPath=FILE . . . . .	8
3.2.5	--reconstructedDataPath=FILE . . . . .	8
3.2.6	--postRecolourPath=FILE . . . . .	8
3.2.7	--preInvScalePath=FILE . . . . .	9
3.2.8	--outputBinaryPly=0 1 . . . . .	9
3.2.9	--outputSystem=0 1 . . . . .	9
3.2.10	--outputUnitLength=REAL-VALUE . . . . .	9
3.2.11	--outputPrecisionBits=INT-VALUE . . . . .	9
3.2.12	--convertPlyColourspace=0 1 . . . . .	9
<b>4</b>	<b>Decoder-specific options</b>	<b>9</b>
4.0.1	--skipOctreeLayers=INT-VALUE . . . . .	9
4.0.2	--decodeMaxPoints=INT-VALUE . . . . .	9

4.0.3	--decoderMemoryManagementMethod=0 1	10
<b>5</b>	<b>Encoder-specific options</b>	<b>10</b>
5.1	Co-ordinate systems and pre-scaling	10
5.1.1	--srcUnit=0 1 metre	10
5.1.2	--srcUnitLength=REAL-VALUE	10
5.1.3	--inputScale=REAL-VALUE	10
5.1.4	--codingScale=REAL-VALUE	10
5.1.5	--sequenceScale=REAL-VALUE	10
5.1.6	--externalScale=REAL-VALUE	10
5.1.7	--autoSeqBbox=0 1	10
5.1.8	--seqOrigin=x,y,z	11
5.1.9	--seqSizeWhd=w,h,d	11
5.1.10	--mergeDuplicatedPoints=0 1	11
5.1.11	--sortInputByAzimuth=0 1	11
5.2	Input partitioning (slices & tiles)	11
5.2.1	--partitionMethod=0 2 3 4 5	11
5.2.2	--partitionNumUniformGeom=INT-VALUE	11
5.2.3	--partitionOctreeDepth=INT-VALUE	11
5.2.4	--sliceMaxPointsTrisoup=INT-VALUE	11
5.2.5	--sliceMaxPoints=INT-VALUE	12
5.2.6	--sliceMinPoints=INT-VALUE	12
5.2.7	--tileSize=INT-VALUE	12
5.2.8	--safeTrisoupPartitioning=0 1	12
5.2.9	--improvedSplitDirection=0 1	12
5.3	General options	12
5.3.1	--geometry_axis_order=INT-VALUE	12
5.3.2	--InterEntropyContinuationEnabled=0 1	12
5.3.3	--disableAttributeCoding=0 1	12
5.3.4	--enforceLevelLimits=0 1	12
5.3.5	--cabac_bypass_stream_enabled_flag=0 1	12
5.3.6	--entropyContinuationEnabled=0 1	12
5.3.7	--bypassBinCodingWithoutProbUpdate=0 1	12
5.4	Geometry coding	13
5.4.1	--positionQuantisationEnabled=0 1	13
5.4.2	--geomTreeType=0 1	13
5.4.3	--positionQuantisationMethod=0 1 2	13
5.4.4	--positionBaseQp=INT-VALUE	13
5.4.5	--positionQpMultiplierLog2=0 1 2 3	13
5.4.6	--positionIdcmQp=INT-VALUE	13
5.4.7	--positionSliceQpOffset=INT-VALUE	13
5.4.8	--positionQuantisationOctreeDepth=INT-VALUE	13
5.4.9	--positionQuantisationSizeLog2=INT-VALUE	13
5.4.10	--qtbEnabled=0 1	13
5.4.11	--maxNumQtBtBeforeOt=INT-VALUE	14
5.4.12	--minQtbtSizeLog2=INT-VALUE	14
5.4.13	--bitwiseOccupancyCoding=0 1	14
5.4.14	--neighbourAvailBoundaryLog2=INT-VALUE	14
5.4.15	--inferredDirectCodingMode=0 1 2 3	14
5.4.16	--jointTwoPointIdcm=0 1	14
5.4.17	--adjacentChildContextualization=0 1	14
5.4.18	--intra_pred_max_node_size_log2=INT-VALUE	14

5.4.19	--planarEnabled=0 1	14
5.4.20	--planarModeIdcmUse=0--32	14
5.4.21	--planarModeThreshold0=0--127	14
5.4.22	--planarModeThreshold1=0--127	14
5.4.23	--planarModeThreshold2=0--127	14
5.4.24	--angularEnabled=0 1	14
5.4.25	--lidarHeadPosition=x,y,z	15
5.4.26	--numLasers=0--255	15
5.4.27	--lasersTheta=FLOAT-VALUE-LIST	15
5.4.28	--lasersZ=FLOAT-VALUE-LIST	15
5.4.29	--lasersNumPhiPerTurn=INT-VALUE-LIST	15
5.4.30	--planarBufferDisabled=0 1	15
5.4.31	--numOctreeEntropyStreams=INT-VALUE	15
5.4.32	--trisoupNodeSizeLog2=INT-VALUE INT-VALUE-LIST	15
5.4.33	--trisoup_sampling_value=INT-VALUE	15
5.4.34	--trisoupQuantizationBits=INT-VALUE INT-VALUE-LIST	15
5.4.35	--trisoupCentroidResidualEnabled=0 1	16
5.4.36	--trisoupHaloEnabled=0 1	16
5.4.37	--trisoupAdaptiveHaloEnabled=0 1	16
5.4.38	--trisoupFineRayTracingEnabled=0 1	16
5.4.39	--trisoupNonCubicNodeNearOriginSideEnabled=0 1	16
5.4.40	--trisoupNonCubicNodeFarFromOriginSideEnabled=0 1	16
5.4.41	--trisoupImprovedEncoderEnabled=0 1	16
5.4.42	--predGeomSort=INT-VALUE	16
5.4.43	--predGeomAzimuthSortPrecision=INT-VALUE	16
5.4.44	--predGeomTreePtsMax=INT-VALUE	16
5.4.45	--positionBaseQpFreqLog2=INT-VALUE	16
5.4.46	--positionSliceQpFreqLog2=INT-VALUE	16
5.4.47	--positionAzimuthScaleLog2=INT-VALUE	16
5.4.48	--positionRadiusInvScaleLog2=INT-VALUE	16
5.4.49	--positionAzimuthSpeed=INT-VALUE	16
5.4.50	--predGeomAzimuthQuantization=0 1	16
5.4.51	--pointCountMetadata=0 1	17
5.4.52	--octreeDepthPlanarEligibilityEnabled=0 1	17
5.4.53	--octreePlanarDynamicOBUFEligibilityEnabled=0 1	17
5.4.54	--multiplePlanarEnabled=0 1	17
5.4.55	--octreeAngularExtension=0 1	17
5.4.56	--disable_planar_IDCM_angular=0 1	17
5.4.57	--interAzimScaleLog2=INT-VALUE	17
5.4.58	--randomAccessPeriod=INT-VALUE	17
5.4.59	--interPredictionEnabled=0 1	17
5.4.60	--biPredictionEnabled=0 1 2	17
5.4.61	--frameMergeEnabled=0 1	17
5.4.62	--predictionPeriod=INT-VALUE	17
5.4.63	--downsamplingRange=INT-VALUE	17
5.4.64	--globalMotionEnabled=0 1	17
5.4.65	--motionVectorPath=FILE	17
5.4.66	--lpuType=0 1	17
5.4.67	--globalMotionBlockSize=w,h,d	17
5.4.68	--globalMotionWindowSize=INT-VALUE	17
5.4.69	--globalMotionSrcType=0 1 2	17

5.4.70	--deriveGMThreshold=0 1	18
5.4.71	--gmThresholdHistScale=0 1	18
5.4.72	--gmThresholdMinZ=0 1	18
5.4.73	--gmThresholdMaxZ=0 1	18
5.4.74	--gmThresholdLeftScale=0 1	18
5.4.75	--gmThresholdRightScale=0 1	18
5.4.76	--use_cuboidal_regions_in_GM_estimation=0 1	18
5.4.77	--predGeomMaxPredIdxTested=0 1	18
5.4.78	--predGeomRadiusPredThreshold=0 1	18
5.4.79	--zCompensationEnabled=0 1	18
5.4.80	--xyCompensationEnabled=0 1	18
5.4.81	--xyCompensationPrePtsNumMinus1=INT-VALUE	18
5.4.82	--secondaryResidualDisabled=0 1	18
5.4.83	--enableGroundPartition=0 1	18
5.4.84	--resRContextQphiThreshold=INT-VALUE	18
5.4.85	--resRContextQphiThresholdPresentFlag=0 1	19
5.4.86	--nodeUniqueDSE=0 1	19
5.4.87	--trisoupFaceVertexEnabled=0 1	19
5.4.88	--interIDCMPredEnabled=0 1	19
5.4.89	--one_point_alone_laser_beam_flag=0 1	19
5.4.90	--paddingForTrisoupPartitionning=0 1	19
5.4.91	--maxPointsPerEntryMinus1=INT-VALUE	19
5.4.92	--trisoupVertexMerge=0 1	19
5.4.93	--trisoupVertexFix=0 1	19
5.5	Attribute coding	19
5.5.1	--attribute=NAME	19
5.5.2	--defaultValue=INT-VALUE-LIST	19
5.5.3	--rahtExtension=0 1	19
5.5.4	--integerHaar=0 1	19
5.5.5	--bitdepth=INT-VALUE	19
5.5.6	--colourMatrix=INT-VALUE	19
5.5.7	--attrScale=INT-VALUE ` and `--attrOffset=INT-VALUE	20
5.5.8	--transformType=0 1 2	20
5.5.9	--rahtPredictionEnabled=0 1	20
5.5.10	--rahtPredictionThreshold0=0--19	20
5.5.11	--rahtPredictionThreshold1=0--19	20
5.5.12	--rahtSubnodePredictionEnabled=0 1	20
5.5.13	--rahtPredictionSearchRange=INT-VALUE	20
5.5.14	--rahtPredictionWeights=INT-VALUE-LIST	20
5.5.15	--numberOfNearestNeighboursInPrediction=INT-VALUE	20
5.5.16	--adaptivePredictionThreshold=INT-VALUE	20
5.5.17	--predWeightBlending=0 1	21
5.5.18	--direct_avg_predictor_disabled_flag=0 1	21
5.5.19	--interComponentPredictionEnabled=0 1	21
5.5.20	--lastComponentPredictionEnabled=0 1	21
5.5.21	--intraLoDSearchRange=INT-VALUE	21
5.5.22	--interLoDSearchRange=INT-VALUE	21
5.5.23	--max_num_direct_predictors=INT-VALUE	21
5.5.24	--lodDecimator=0 1 2	21
5.5.25	--intraLodPredictionSkipLayers=INT-VALUE	21
5.5.26	--aps_scalable_enabled_flag=0 1	21

5.5.27	--max_neigh_range=INT-VALUE	21
5.5.28	--levelOfDetailCount=INT-VALUE	21
5.5.29	--dist2=INT-VALUE	21
5.5.30	--dist2PercentileEstimate=FLOAT-VALUE	22
5.5.31	--positionQuantizationScaleAdjustsDist2=0 1	22
5.5.32	--lodSubsamplingPeriod=INT-VALUE INT-VALUE-LIST	22
5.5.33	--canonical_point_order_flag=0 1	22
5.5.34	--spherical_coord_flag=0 1	22
5.5.35	--attrSphericalMaxLog2=INT-VALUE	22
5.5.36	--enableMortonCodeScaling=0 1	22
5.5.37	--lod_neigh_bias=INT-VALUE-LIST	22
5.5.38	--qp=INT-VALUE	22
5.5.39	--qpChromaOffset=INT-VALUE	22
5.5.40	--aps_slice_qp_deltas_present_flag=0 1	22
5.5.41	--qpLayerOffsetsLuma=INT-VALUE-LIST	22
5.5.42	--qpLayerOffsetsChroma=INT-VALUE-LIST	22
5.5.43	--quantNeighWeight=INT-VALUE-LIST	22
5.5.44	--predictionWithDistributionEnabled=0 1	23
5.5.45	--max_points_per_sort_log2_plus1=INT-VALUE	23
5.5.46	--attributeInterPredictionEnabled=0 1	23
5.5.47	--attrInterPredSearchRange=INT-VALUE	23
5.5.48	--rahtEnableCodeLayer=0 1	23
5.5.49	--rahtEnableIntraPredCodeLayer=0 1	23
5.5.50	--rahtInterPredictionDepthMinus1=INT-VALUE	23
5.5.51	--attrInterPredTranslationThresh=INT-VALUE	23
5.5.52	--rahtInterSendFilters=0 1	23
5.5.53	--rahtInterSkipFilteringLayers=INT-VALUE	23
5.5.54	--rahtInterEligibilityLowerTh=INT-VALUE	23
5.5.55	--rahtInterEligibilityUpperTh=INT-VALUE	23
5.5.56	--rahtHybridPredictionEnabled=0 1	23
5.5.57	--rahtHybridPredictionEnabledMinLayerMinus1=INT-VALUE	23
5.5.58	--rahtHybridPredictionEnabledLayersCount=INT-VALUE	23
5.5.59	--QPShiftStep=INT-VALUE	23
5.5.60	--attrInterIntraSliceRDO=0 1	23
5.5.61	--cross_attr_prediction_enabled_flag=0 1	23
5.5.62	--attrMultiEncodedOrder=0 1	23
5.5.63	--cross_attr_prediction_enabled_this_type=0 1	23
5.5.64	--refAttrIdx=INT-VALUE	23
5.5.65	--alphaY=FLOAT-VALUE	24
5.5.66	--alphaCb=FLOAT-VALUE	24
5.5.67	--alphaCr=FLOAT-VALUE	24
5.5.68	--alphaZeroTriplet=FLOAT-VALUE	24
5.5.69	--alphaRefl=DOUBLE-VALUE	24
5.5.70	--alphaZeroTripletRefl=DOUBLE-VALUE	24
5.5.71	--chromaFormat420=0 1	24
5.6	Attribute recolouring (encoder only)	24
5.6.1	--recolourSearchRange=INT-VALUE	24
5.6.2	--recolourNumNeighboursFwd=INT-VALUE	24
5.6.3	--recolourNumNeighboursBwd=INT-VALUE	24
5.6.4	--recolourUseDistWeightedAvgFwd=0 1	24
5.6.5	--recolourUseDistWeightedAvgBwd=0 1	24

5.6.6	--recolourSkipAvgIfIdenticalSourcePointPresentFwd=0 1 . . .	24
5.6.7	--recolourSkipAvgIfIdenticalSourcePointPresentBwd=0 1 . . .	24
5.6.8	--recolourDistOffsetFwd=REAL-VALUE . . . . .	24
5.6.9	--recolourDistOffsetBwd=REAL-VALUE . . . . .	24
5.6.10	--recolourMaxGeometryDist2Fwd=REAL-VALUE . . . . .	24
5.6.11	--recolourMaxGeometryDist2Bwd=REAL-VALUE . . . . .	24
5.6.12	--recolourMaxAttributeDist2Fwd=REAL-VALUE . . . . .	24
5.6.13	--recolourMaxAttributeDist2Bwd=REAL-VALUE . . . . .	24
5.7	layer-group slicing . . . . .	24
5.7.1	--layerGroupEnabledFlag=0 1 . . . . .	24
5.7.2	--numLayerGroupsMinus1=INT-VALUE . . . . .	25
5.7.3	--numLayersPerLayerGroup=INT-VALUE INT-VALUE-LIST . . . . .	25
5.7.4	--subgroupBboxSize_Cubic=INT-VALUE . . . . .	25
5.7.5	--enableSliceSelector=0 1 2 . . . . .	25
5.7.6	--roiEnabledFlag=0 1 . . . . .	25
5.7.7	--roiSize=INT-VALUE-LIST . . . . .	25
5.7.8	--bboxScaleForRoiPoint=FLOAT-VALUE-LIST . . . . .	25
5.7.9	--numSkipLayerGroups=INT-VALUE . . . . .	25
5.7.10	--rootLayerGroupContextReferenceFlag=0 1 . . . . .	25
5.7.11	--depth1stSubgroupSearch=0 1 . . . . .	25
5.7.12	--attributeContextReferenceIDPresentFlag=0 1 . . . . .	25
5.7.13	--qpCoefDependentUnits=INT-VALUE . . . . .	25
5.7.14	--qpLayerCoefDependentUnits=INT-VALUE . . . . .	25
5.7.15	--weightAdjustmentEnabledFlag=0 1 . . . . .	25
5.7.16	--weightAdjustmentMethod=0 1 . . . . .	25
5.7.17	--geomAttrInterleavingEnabled=0 1 . . . . .	25

## General Information

Reference software is being made available to provide a reference implementation of the G-PCC standard being developed by MPEG 3DGH (ISO/IEC JTC1/SC29/WG7). One of the main goals of the reference software is to provide a basis upon which to conduct experiments in order to determine which coding tools provide desired coding performance. It is not meant to be a particularly efficient implementation of anything, and one may notice its apparent unsuitability for a particular use. It should not be construed to be a reflection of how complex a production-quality implementation of a future G-PCC standard would be.

This document aims to provide guidance on the usage of the reference software. It is widely suspected to be incomplete and suggestions for improvements are welcome. Such suggestions and general inquiries may be sent to the general MPEG 3DGH email reflector at [mpeg-3dgc@gti.ssr.upm.es](mailto:mpeg-3dgc@gti.ssr.upm.es) (registration required).

## Bug reporting

Bugs should be reported on the issue tracker set up at <https://git.mpeg.expert/MPEG/3dgh/g-pcc/software/tm/mpeg-pcc-tmc13/-/issues>.

## Obtaining the software

The authoritative location of the software is the following git repository:

<https://git.mpeg.expert/MPEG/3dgh/g-pcc/software/tm/mpeg-pcc-tmc13>

Each released version may be identified by a version control system tag in the form ‘release-v\$version’.

An example (release version v29.0):

```
$ git clone https://git.mpeg.expert/MPEG/3dgh/g-pcc/software/tm/mpeg-pcc-tmc13.git
$ cd mpeg-pcc-tmc13
$ git checkout release-v29.0
```

It is strongly advised to obtain the software using the version control system rather than to download a zip (or other archive) of a particular release. The build system uses the version control system to accurately identify the version being built.

## **Building**

The codec is supported on Linux, OSX and Windows platforms. The build configuration is managed using CMake.

It is strongly advised to build the software in a separate build directory.

### **Linux**

```
$ mkdir build
$ cd build
$ cmake ..
$ make
$ tmc3/tmc3 --help
```

### **OSX**

```
$ mkdir build
$ cd build
$ cmake .. -G Xcode
$ xcodebuild
$ tmc3/tmc3 --help
```

As an alternative, the generated XCode project may be opened and built from XCode itself.

### **Windows**

```
> md build
> cd build
> cmake .. -G "Visual Studio 15 2017 Win64"
```

Open the generated visual studio solution to build it.

## **Using the codec**

```
./tmc3 [-help] [-c config.cfg] [--parameter=value]
```

The encoder takes as input one or more PLY files describing a point cloud sequence with integer positions and, optionally, per-point integer colour and reflectance attributes.

The output of the encoder is a binary bitstream encapsulated using the G-PCC annex-B format.



Conversely, the decoder takes as input a compressed bitstream file in G-PCC annex-B format and produces one or more reconstructed PLY file with position and any present attribute values.

The software may be configured using either command line arguments or from a configuration file specified using the `'-c|--config='` option.

Sample configuration files are provided in the `cfg/` directory. The utility `<scripts/gen-cfg.sh>` may be used to generate per sequence and per rate point configuration files for a variety of common test conditions.

Parameters are set by the last value encountered on the command line. Therefore if a setting is set via a configuration file, and then a subsequent command line parameter changes that same setting, the command line parameter value will be used.

## General options

### **--help**

Print a list of available command line (and configuration file) options along with their default values and exit.

### **--config=FILE, -c**

This specifies a configuration file to be immediately loaded.

### **--mode=VALUE**

This option selects the codec's mode of operation. A value of 0 enables encoding functionality. A value of 1 switches to decoding mode.

## I/O parameters

### **--firstFrameNum=INT-VALUE**

The initial frame number of the input or output sequence. The software replaces any instance of a `'%d'` printf format directive with the current frame number when evaluating the following options:

- `uncompressedDataPath` - `reconstructedDataPath` - `postRecolourPath` - `preInvScalePath`

NB: When decoding, this option relates only to the output file names.

In order to have the decoder produce identically numbered output ply files as the encoder input, specify the same value of `firstFrameNum` for the decoder.

### **--frameCount=INT-VALUE**

(Encoder only) The number of frames to be encoded.

### **--uncompressedDataPath=FILE**

(Encoder only) The input source point cloud to be compressed. The first instance of `'%d'` in FILE will be expanded with the current frame number.

### **--compressedStreamPath=FILE**

The compressed bitstream file output when encoding or input when decoding.

### **--reconstructedDataPath=FILE**

The reconstructed point cloud file. When encoding, the output is the locally decoded picture. It is expected that the reconstructed output of the encoder and decoder match exactly.

The first instance of `'%d'` in FILE will be expanded with the current frame number.

### **--postRecolourPath=FILE**

(Encoder only) As part of the encoding process, it may be necessary to re-colour the point cloud if the point geometry is altered. This diagnostic output file corresponds to the re-coloured point cloud prior to attribute coding without output geometry scaling.

The first instance of `'%d'` in FILE will be expanded with the current frame number.

### **--preInvScalePath=FILE**

(Decoder only) This diagnostic output corresponds to the decoded point cloud (geometry and attributes) prior to output geometry scaling.

When compared to the output of 'postRecolourPath', the performance of attribute coding may be directly measured without being confounded by any geometry losses.

The first instance of '%d' in FILE will be expanded with the current frame number.

### **--outputBinaryPly=0|1**

Sets the output format of PLY files (Binary=1, ASCII=0). Reading and writing binary PLY files is more efficient than the ASCII variant, but are less suited to simple scripts and direct human inspection.

If outputting non-integer point co-ordinates (eg, due to the output geometry scaling), the precision of the binary and ASCII versions are not identical.

### **--outputSystem=0|1**

Controls the output scaling of the coded point cloud.

Value	Description
0	Conformance output
1	External co-ordinate system

The conformance output scales the coded point cloud to the sequence co-ordinate system. The output point positions are not offset by the sequence origin.

The external co-ordinate system output scales the point cloud to the defined external co-ordinate system (see 'sequenceScale', 'externalScale', and 'outputUnitLength'). The output point positions are offset by the sequence origin, appropriately scaled.

### **--outputUnitLength=REAL-VALUE**

The length of the output point cloud unit vector. Point clouds output by the encoder or decoder are rescaled to match this length.

For example, 'outputUnitLength=1000' outputs a point cloud with integer point positions representing millimetres.

### **--outputPrecisionBits=INT-VALUE**

The number of fractional bits to retain when scaling from the coding co-ordinate system to the sequence co-ordinate system. The fractional bits are further retained when converting to the external co-ordinate system.

The special value 'outputPrecisionBits=-1' retains all fractional bits during the scaling process.

### **--convertPlyColourspace=0|1**

Controls the conversion of ply RGB colour attributes to/from the colourspace set by an attribute's 'colourMatrix' before attribute coding and after decoding. When disabled (0), or if there is no converter available for the requested 'colourMatrix', no conversion happens; however the 'colourMatrix' value is still written to the bitstream.

## **Decoder-specific options**

### **--skipOctreeLayers=INT-VALUE**

The option indicates the number of skipped lod layers from leaf lod. If `aps.scalable_enable_flag` is 1, the option is valid. Otherwise, the option is ignored.

### **--decodeMaxPoints=INT-VALUE**

A value greater than zero controls the automatic derivation of 'skipOctreeLayers' such that at most  $n$  points are decoded. This option only has an effect if the bitstream contains per octree level point count metadata (see 'pointCountMetadata').

**--decoderMemoryManagementMethod=0|1**

When the value is 0, decoder memory management uses num\_subsequent\_subgroups, and when the value is 1, it uses the layer-group structure inventory.

## Encoder-specific options

### Co-ordinate systems and pre-scaling

**--srcUnit=0|1|metre**

The physical unit used to interpret values of 'srcUnitLength'.

Value	Description
0	dimensionless
1,metre	metre

**--srcUnitLength=REAL-VALUE**

The length of the source point cloud unit vector. This value is used to define the unit vector length of the sequence co-ordinate system. It is not used to perform scaling by the encoder.

For example, 'srcUnitLength=1000' and 'srcUnit=metre' indicates that integer positions in the source point cloud represent millimetres.

**--inputScale=REAL-VALUE**

A scale factor applied to point positions in the source point cloud prior to integer conversion. The 'inputScale' changes the length of the source unit vectors (as set by 'srcUnitLength').

For example, a point cloud may have a unit vector representing 1 metre ('srcUnitLength=1') and contain points with a resolution of 1000 points per metre. Since the codec can only represent integer positions, without input scaling, it is coded with a precision of one metre. Setting 'inputScale=1000' will increase the precision to 1 millimetre.

**--codingScale=REAL-VALUE**

A scale factor used to determine the length of the coding co-ordinate system unit vector. The scale factor is relative to 'inputScale'. The input point cloud (after integer conversion) is scaled by 'codingScale' and rounded to integer positions.

If 'codingScale' is greater than 'sequenceScale', the encoder will set 'codingScale=sequenceScale'.

A decoder will scale the coded point cloud by 'sequenceScale/codingScale' prior to output.

**--sequenceScale=REAL-VALUE**

A scale factor used to determine the length of the sequence co-ordinate system unit vector. The scale factor is relative to 'inputScale'. The input point cloud (after integer conversion) is scaled by the smallest of 'sequenceScale' and 'codingScale'.

**--externalScale=REAL-VALUE**

A scale factor used to define the length of the sequence co-ordinate system when 'srcUnit' is dimensionless. The scale factor is relative to 'inputScale'. The 'externalScale' does not affect scaling of the input point cloud prior to coding.

For example, a point cloud coded with 'sequenceScale=0.25' and 'externalScale=0.5' specifies that:

- the input is scaled by 0.25 prior to coding, and - the decoder is informed that 1 sequence unit is equal to 2 external units.

NB: a decoder is not required to scale the sequence co-ordinate system to an external co-ordinate system prior to output.

**--autoSeqBbox=0|1**

Automatically determine the sequence bounding box ('seqOrigin' and 'seqSizeWhd') using the first input frame.

**--seqOrigin=x, y, z**

Sets the origin of the sequence bounding box. The 'seqOrigin' must be less than or equal to the lowest input point position. The origin is configured in the input co-ordinate system (after integer conversion). The encoder will adjust the origin according to 'sequenceScale'.

This option has no effect when 'autoSeqBbox=1'.

**--seqSizeWhd=w, h, d**

Sets the size of the sequence bounding box. The size is configured in the input co-ordinate system (after integer conversion). The encoder will adjust the size according to 'sequenceScale'.

'seqSizeWhd=0,0,0' disables signalling the sequence bounding box size.

This option has no effect when 'autoSeqBbox=1'.

**--mergeDuplicatedPoints=0 | 1**

Controls the ability to code duplicate points. When duplicate point merging is enabled, bitstream syntax related to duplicate points is disabled and a pre-filtering process is used to remove co-located points.

**--sortInputByAzimuth=0 | 1**

Pre-sort the input point cloud according to azimuth angle with the origin 'lidarHeadPosition'. Pre-sorting occurs prior to tile/slice partitioning.

## Input partitioning (slices & tiles)

**--partitionMethod=0 | 2 | 3 | 4 | 5**

Selects the partitioning method to map points to tiles and slices:

Value	Description
0	none (single slice)
2	uniform partitioning along longest edge
3	uniform octree partitions
4	uniform square partitions
5	n-point spans

Uniform longest edge partitioning slices the point cloud along the longest edge according to 'partitionNumUniformGeom'.

Uniform octree partitioning generates slices with the same size based on an octree partitioning of the point cloud according to 'partitionOctreeDepth'.

Uniform square partitioning generates cubic slices sized according to the shortest edge.

N-point span partitioning divides the input point list (after input pre-sorting) into 'sliceMaxPoints'-point sublists. Input order (after pre-sorting) is maintained.

In all cases, a refinement process may merge or split slices in order to satisfy maximum or minimum points per slice constraints.

**--partitionNumUniformGeom=INT-VALUE**

Sets the number of slices to generate using 'partitionMethod=2'. If equal to zero, the number of slices is the integer ratio of the longest to shortest edges of the point cloud bounding box.

**--partitionOctreeDepth=INT-VALUE**

Sets the depth of the octree for slice generation using 'partitionMethod=3'.

The input point cloud is decomposed using an octree with the configured depth. Each occupied leaf of the octree represents a single slice.

**--sliceMaxPointsTrisoup=INT-VALUE**

Upper limit to the number of reconstructed points in each slice with trisoup. Trisoup reconstruction is subsampled until this constraint is satisfied.

**--sliceMaxPoints=INT-VALUE**

Upper limit to the number of in each slice. Slices are split until this constraint is satisfied.

**--sliceMinPoints=INT-VALUE**

Minimum number of points in each slice. This soft limit is used to merge small slices together.

**--tileSize=INT-VALUE**

Tile dimension to use when performing initial partitioning. A value of zero disables tile partitioning.

**--safeTrisoupPartitioning=0|1**

When uniform partitioning along longest edge or uniform square partitioning is used additionnal constraints are added on section boundaries to ensure not splitting withing Trisoup nodes. This may result in slices containing more than 'sliceMaxPoints' points, and/or in a not perfectly uniform partitioning.

**--improvedSplitDirection=0|1**

(Encoder only) Controls the activation of the improved slice split direction determination after the uniform square partitioning. The activation of the method can minimize the number of duplicate vertices on slice boundaries.

## General options

**--geometry\_axis\_order=INT-VALUE**

Configures the order in which axes are internally coded. Changing the axis order does not change the orientation of the reconstructed point cloud.

Value	Coding order
0	z, y, x
1	x, y, z
2	x, z, y
3	y, z, x
4	z, y, x
5	z, x, y
6	y, x, z
7	x, y, z

**--InterEntropyContinuationEnabled=0|1**

Propagate context state for P frame from the preceding frame.

**--disableAttributeCoding=0|1**

This option instructs the encoder to ignore all options relating to attribute coding, as if they had never been configured.

**--enforceLevelLimits=0|1**

Controls the enforcement of level limits by the encoder. If a level limit is violated, the encoder will abort.

**--cabac\_bypass\_stream\_enabled\_flag=0|1**

Controls the entropy coding method used for equi-probable (bypass) bins:

Value	Description
0	bypass bins coded using CABAC
1	bypass bins coded in bypass substream

**--entropyContinuationEnabled=0|1**

Controls the propagation of entropy coding state (context values) between slices in the same frame. When enabled, each slice (except the first) has a coding dependency on the previous slice.

**--bypassBinCodingWithoutProbUpdate=0|1**

Controls the coding of bypass bins without any probability update; this enables to reduce some complexity in coding the bypass bins.

## Geometry coding

**--positionQuantisationEnabled=0|1**

Enables in-loop quantisation and reconstruction of geometry positions.

NB: All in-loop quantisation is independent (and happens after) any position scaling due to 'positionQuantizationScale'.

**--geomTreeType=0|1**

Selects the geometry tree coding method.

Value	Description
0	Octree
1	Predictive geometry tree

**--positionQuantisationMethod=0|1|2**

Selects the method used to determine the QP value for each quantised tree node.

Value	Description
0	Uniform
1	Random
2	By point density

The 'uniform' method sets every node QP to the slice QP.

The 'random' method picks a uniformly distributed random QP for each node from the range of permitted values. The seed for random number generation may be set using the environment variable 'SEED'.

The 'point density' method varies the per-node qp according to the relative number of points in each node. The sparsest 5% of nodes use  $\text{sliceQp} + \text{qpPot}$ , the densest 40% of nodes use  $\text{sliceQp} - \text{qpPot}$ , and the remaining nodes use  $\text{sliceQp}$ , where  $\text{qpPot}$  is ' $8 >> \text{positionQpMultiplierLog2}$ '.

**--positionBaseQp=INT-VALUE**

The quantisation parameter used to quantise geometry positions. The effective QP may be varied according to 'positionSliceQpOffset' and 'positionQuantisationOctreeDepth'. A QP equal to 0 results in a scale factor of 1.

**--positionQpMultiplierLog2=0|1|2|3**

Controls the granularity of quantisation step sizes by limiting the number of QP values per step size doubling interval. There are  $2^n$  QPs per step size doubling interval.

**--positionIdcmQp=INT-VALUE**

The quantisation parameter used to quantise directly coded (IDCM) point positions prior to reaching the 'positionQuantisationOctreeDepth'.

**--positionSliceQpOffset=INT-VALUE**

A per-slice offset to be applied to 'positionBaseQp'.

**--positionQuantisationOctreeDepth=INT-VALUE**

The depth in the octree at which per-node QP offsets are signalled. A non-normative encoder process determines the QP offset based upon the local density of the octree. A value of -1 disables signalling of per-node QP offsets.

**--positionQuantisationSizeLog2=INT-VALUE**

Sets the depth at which per-node QP offsets are signalled. The depth is the tree level with the configured node size. This value, if greater than 0, overrides 'positionQuantisationOctreeDepth'.

When non-cubic nodes are present, the depth is the tree level with the minimum node size dimension.

**--qtbEnabled=0|1**

Enables non-cubic geometry tree coding. When enabled, the geometry tree may have a cuboid bounding box. The partitioning of internal tree nodes at a particular depth are determined non-normatively by the encoder to be one of octree, quadtree or binary partitions.

**--maxNumQtBtBeforeOt=INT-VALUE**

Limits the maximal number of quadtree and binary tree partitions used before the first octree partition.

**--minQtbtSizeLog2=INT-VALUE**

Specifies the minimum size of quadtree and binary tree partitions.

**--bitwiseOccupancyCoding=0|1**

In octree geometry coding, there are both byte-wise and bit-wise tools to encode the occupancy data. This option selects between the two methods.

**--neighbourAvailBoundaryLog2=INT-VALUE**

Defines the volume within which octree nodes are considered available for use in occupancy contextualisation and intra occupancy prediction.

A value less than 2 limits the use of neighbouring nodes to direct octree siblings.

The software currently supports a maximum value of 8 or 9 when intra occupancy prediction is enabled or disabled respectively.

**--inferredDirectCodingMode=0|1|2|3**

Controls the degree to which early termination of the geometry octree is used to code isolated points.

Value	Extent of qualifying node criteria
0	Disabled
1	Fully isolated parent and child
2	Partially isolated parent
3	Unconstrained

**--jointTwoPointIdcm=0|1**

Controls the method used to code the point positions of directly coded nodes containing two distinct points. When enabled, an implicit point order is used to improve coding efficiency.

**--adjacentChildContextualization=0|1**

Controls the contextualization of occupancy bits according to the state of adjacent children of neighbouring nodes.

**--intra\_pred\_max\_node\_size\_log2=INT-VALUE**

Intra occupancy prediction uses an octree node's neighbours to predict its occupancy. The prediction mode is enabled for octree nodes smaller than or equal to the configured size. A value of 0 disables intra occupancy prediction.

**--planarEnabled=0|1**

Controls the use of planar coding mode for geometry occupancy.

**--planarModeIdcmUse=0--32**

Controls the frequency in 1/32 percent of IDCM eligibility. Set to zero, IDCM is disabled. Set to 32, IDCM is unconstrained.

**--planarModeThreshold0=0--127**

Controls the eligibility threshold of the first planar mode based upon local child node density when the eligibility is not determined based on octree depth.

**--planarModeThreshold1=0--127**

Controls the eligibility threshold of the second planar mode based upon local child node density when the eligibility is not determined based on octree depth.

**--planarModeThreshold2=0--127**

Controls the eligibility threshold of the third planar mode based upon local child node density when the eligibility is not determined based on octree depth.

**--angularEnabled=0|1**

Controls the use of the angular coding mode in geometry occupancy contextualisation. Angular coding mode uses a LiDaR head model prior to improve the compression of a LiDaR acquired point cloud. The angular mode requires that planar mode is enabled.

**--lidarHeadPosition=x,y,z**

Specifies the LiDAR head position for use by the angular mode in terms of the input (unquantised) point cloud co-ordinate system.

**--numLasers=0--255**

The number of known laser angles and positions for use in angular mode.

**--lasersTheta=FLOAT-VALUE-LIST**

The elevation angle, theta, of each known laser used by the angular mode. Each elevation angle is expressed in radians relative to the x-y plane (range:  $[-\pi, \pi]$ ). The zero angle describes a horizontal laser. Positive angles represent an elevation above the horizontal. Negative angles represent an elevation below the horizontal.

The software will convert the floating point angles to an 18 bit fixed point representation.

**--lasersZ=FLOAT-VALUE-LIST**

The vertical offset of each known laser used by the angular mode. Each offset is expressed along the z axis in the input point cloud co-ordinate system, corresponding to a vertical offset relative to the LiDAR head position.

The software will convert the floating point offsets to a three bit fixed point representation.

**--lasersNumPhiPerTurn=INT-VALUE-LIST**

The maximum number of samples that can be acquired during a full rotation of each known laser used by the angular mode.

**--planarBufferDisabled=0|1**

Controls the deactivation of the planar mode buffer for angular mode.

**--numOctreeEntropyStreams=INT-VALUE**

The number of geometry sub-streams (suitable for parallel coding) used to encode the geometry octree. For example, a value of eight generates eight sub-streams, one for the initial tree, then one for each of the last seven tree levels.

No parallel sub-streams are generated when \*VALUE\* is 1.

NB: the reference software does not attempt to exploit any opportunities for parallelism generated by this feature.

**--trisoupNodeSizeLog2=INT-VALUE|INT-VALUE-LIST**

Controls the use of trisoup by setting the node size for triangle based surface reconstruction. The trisoup method terminates the octree coding at the given node size and continues by encoding triangles which are subsequently voxelised to produce points.

A value less than 2 disables the use of trisoup.

When a list of values is used, the n-th entry in the list controls the configuration of the n-th slice. The last entry is mapped to all remaining slices.

**--trisoup\_sampling\_value=INT-VALUE**

Controls the number of points generated by the trisoup triangle voxelisation process. Larger values reduce the number of points generated per triangle.

Value	Description
0	automatic (default)
1	no sub-sampling
N >=1	point sampling period

The automatic mode will find the smallest sampling value that such that the number of generated points does not exceed the slice limit set by 'sliceMaxPoints'.

**--trisoupQuantizationBits=INT-VALUE|INT-VALUE-LIST**

Number of bits used for quantization of position of trisoup vertices along edges.



**--trisoupCentroidResidualEnabled=0|1**

Controls the activation of coding residual position value for centroid vertex in trisoup coding.

**--trisoupHaloEnabled=0|1**

Controls the activation of using halo around trisoup triangles for ray tracing.

**--trisoupAdaptiveHaloEnabled=0|1**

Controls the activation of using adaptive halo when using halo around trisoup triangles for ray tracing.

**--trisoupFineRayTracingEnabled=0|1**

Controls the activation of additional ray tracing from non-integer origin.

**--trisoupNonCubicNodeNearOriginSideEnabled=0|1**

Controls the activation of the non cubic nodes near the origin side of the slice bounding box. When enabled, the encoder will determine the bounding box origin to be used for the TriSoup volume in each slice.

**--trisoupNonCubicNodeFarFromOriginSideEnabled=0|1**

Controls the activation of the non cubic nodes far from the origin side of the slice bounding box. When enabled, the encoder will determine the bounding size to be used for the TriSoup volume in each slice.

**--trisoupImprovedEncoderEnabled=0|1**

(Encoder only) Controls the activation of improved determination of trisoup vertex position.

**--predGeomSort=INT-VALUE**

Point order used to construct predictive geometry trees. Requires 'geomTreeType=1'.

Value	Description
0	none
1	morton order
2	azimuth angle
3	radial distance
4	source azimuth angle (ply: laserangle)

**--predGeomAzimuthSortPrecision=INT-VALUE**

Controls the precision used in azimuthal sorting of points prior to predictive tree construction. A value of 0 represents full-precision, otherwise larger values represent increasing precision. Requires 'predGeomSort=2'.

**--predGeomTreePtsMax=INT-VALUE**

Maximum number of points per predictive geometry tree. A slice may contain more than one predictive geometry tree. Requires 'geomTreeType=1'.

**--positionBaseQpFreqLog2=INT-VALUE**

Controls the number of predictive geometry tree nodes scaled by the same QP offset instance. QP offsets are signalled every  $2^n$  nodes in tree traversal order. This configuration applies to all slices. Requires 'positionQuantisationEnabled=1'.

**--positionSliceQpFreqLog2=INT-VALUE**

Identical to 'positionBaseQpFreqLog2', but controls per-slice configuration.

**--positionAzimuthScaleLog2=INT-VALUE**

Number of additional bits used to represent predictive geometry azimuth angles. Requires 'angularEnabled=1'.

**--positionRadiusInvScaleLog2=INT-VALUE**

Degree of quantisation applied in the representation of angular predictive geometry radial distances. Requires 'angularEnabled=1'.

**--positionAzimuthSpeed=INT-VALUE**

Step size used to linearly model progression of per-laser azimuthal angles during angular predictive geometry coding. Requires 'angularEnabled=1'.

**--predGeomAzimuthQuantization=0|1**

Controls the use of radius dependent azimuth quantization in predictive geometry coding. Requires 'angularEnabled=1' and 'geomTreeType=1'.

**--pointCountMetadata=0|1**

Controls the addition of per octree layer point count metadata to each geometry slice.

**--octreeDepthPlanarEligibilityEnabled=0|1**

Controls the determination of planar mode eligibility based on octree depth.

**--octreePlanarDynamicOBUFEligibilityEnabled=0|1**

Controls the enabling of Dynamic OBUF in planar mode.

**--multiplePlanarEnabled=0|1|**

Controls the enabling of signalling of planar mode for multiple directions.

**--octreeAngularExtension=0|1**

Controls the enabling of extending angular mode in octree geometry.

**--disable\_planar\_IDCM\_angular=0|1**

Controls the disabling of planar mode for geometry coding of IDCM coded nodes when angular coding is enabled.

**--interAzimScaleLog2=INT-VALUE**

Specifies the scale factor to be applied to azimuth angle during inter search in predictive geometry coding.

**--randomAccessPeriod=INT-VALUE**

Specifies the distance (in frames) between random access points when encoding a sequence.

**--interPredictionEnabled=0|1**

Controls the enabling of inter prediction coding.

**--biPredictionEnabled=0|1|2**

Enable bi-directional inter prediction for octree coding.

Value	Description
0	disable bi-directional inter prediction
1	enable bi-directional inter prediction using IBBB GOF structure
2	enable bi-directional inter prediction using hierarchical GOF

**--frameMergeEnabled=0|1**

Enable frameMerge mode for bi-directional inter prediction.

**--predictionPeriod=INT-VALUE**

Maxium distance (in pictures) between I-frame and P-frame when encoding a sequence using bi-direction inter prediction.

**--downsamplingRange=INT-VALUE**

the downsampling range applied to reference frame: -1: downsampling is disabled; 0: downsampling occurs for points with the same attribute and at the same position; greater than 0: downsampling occurs for points with the same attribute and laser index within the range.

**--globalMotionEnabled=0|1**

Controls the enabling of global motion compensation in inter prediction.

**--motionVectorPath=FILE**

(Encoder only) The source containing the motion vector parameters using in global motion compensation.

**--lpuType=0|1**

Controls the reference points used in motion compensation for LPUs.

Value	Description
0	Use road/object classification-based LPUs
1	Use cuboid partition-based LPUs

**--globalMotionBlockSize=w,h,d**

Specifies the block size used for global motion compensation.

**--globalMotionWindowSize=INT-VALUE**

Specifies the window size used in global motion compensation

**--globalMotionSrcType=0|1|2**

Controls the global motion parameters used for global motion compensation.

Value	Description
0	Use externally specified global motion parameters
1	Use internally derived global motion parameters based on LMS
2	Use internally derived global motion parameters based on ICP

**--deriveGMThreshold=0|1**

(Encoder only) Controls whether the global motion thresholds derived at the encoder.

**--gmThresholdHistScale=0|1**

(Encoder only) Specifies a scale value used to compute histogram in the derivation of global motion compensation thresholds.

**--gmThresholdMinZ=0|1**

(Encoder only) Specifies a maximum z value used to compute histogram in the derivation of global motion compensation thresholds.

**--gmThresholdMaxZ=0|1**

(Encoder only) Specifies a minimum z value used to compute histogram in the derivation of global motion compensation thresholds.

**--gmThresholdLeftScale=0|1**

(Encoder only) Specifies a scale value used to calculate the lower threshold to apply global motion compensation.

**--gmThresholdRightScale=0|1**

(Encoder only) Specifies a scale value used to calculate the upper threshold to apply global motion compensation.

**--use\_cuboidal\_regions\_in\_GM\_estimation=0|1**

(Encoder only) Controls the use of cuboidal regions with square cross-section in xy-plane for global motion estimation using LMS.

Value	Description
0	Use cubic regions
1	Use cuboid regions

**--predGeomMaxPredIdxTested=0|1**

(Encoder only) Specifies the maximum prediction index tested by encoder in prediction list. A value lower than 0 or higher than predGeomMaxPredIdx implies that the maximum prediction index is set equal to predGeomMaxPredIdx. The default value is set to -1.

**--predGeomRadiusPredThreshold=0|1**

Specifies the threshold for consider new predictor in the prediction list for intra prediction in predictive geometry coding. The threshold effectively used is predGeomRadiusPredThreshold scaled by positionRadiusInvScaleLog2.

**--zCompensationEnabled=0|1**

Enables z compensation when using octree coding.

**--xyCompensationEnabled=0|1**

Enables xy compensation when using octree coding.

**--xyCompensationPrePtsNumMinus1=INT-VALUE**

Specifies the number of previous points used when performing xy compensation.

**--secondaryResidualDisabled=0|1**

Controls the disabling of quantized cartesian residual in lossy pred tree geometry coding.

**--enableGroundPartition=0|1**

Controls the enabling of ground/objects partitioning for predictive geometry encoder.

**--resRContextQphiThreshold=INT-VALUE**

Specifies the Qphi threshold used for contextualization of radius residual coding in predictive geometry angular coding.

**--resRContextQphiThresholdPresentFlag=0|1**

Controls the presence of the Qphi threshold used for contextualization of radius residual coding in predictive geometry angular coding.

**--nodeUniqueDSE=0|1**

Enables to calculate a node unique distanceSearchEncoder value, based on node characteristics instead of slice characteristics.

**--trisoupFaceVertexEnabled=0|1**

Controls whether face vertices are activated.

**--interIDCMPredEnabled=0|1**

Controls the eligible of inter idcm coding.

**--one\_point\_alone\_laser\_beam\_flag=0|1**

Indicates whether there is only one point alone one laser beam when interIDCMPredEnabled flag is true.

**--paddingForTrisoupPartitionning=0|1**

(Encoder only) Controls the activation of the slice padding for trisoup.

**--maxPointsPerEntryMinus1=INT-VALUE**

Indicates the maximum number of points per scaled azimuth in the reference frame for predictive geometry coding.

**--trisoupVertexMerge=0|1**

Controls the activation of vertex merge during vertex determination.

**--trisoupVertexFix=0|1**

Controls the activation of vertex fix during vertex determination.

## Attribute coding

The codec may be configured to represent one or more attributes. The configuration of each attribute is independent from all others. To configure coding of an attribute, first set the attribute options, then save the configuration using the ‘attribute’ option.

**--attribute=NAME**

Saves the current attribute configuration for coding the named attribute.

Name	Description
colour	r, g, and b properties as a tri-stimulus attribute
reflectance	refc or reflectance property as a single-stimulus attribute

This option must be specified after the options corresponding to the attribute.

**--defaultValue=INT-VALUE-LIST**

The default value to use for attribute data in case of data loss. If unset, the implicit default attribute value is  $2^{**}(\text{bitdepth}-1)$ .

**--rahtExtension=0|1**

Enables RAHT extension tools including buffee precision increase and skipping transform domain prediction in one-subnode condition.

**--integerHaar=0|1**

Controls the use of Integer Haar Transform method, enabled on under lossless attribute conditions.

**--bitdepth=INT-VALUE**

The bitdepth of the attribute data. NB, this is not necessarily the same as the bitdepth of the PLY property.

**--colourMatrix=INT-VALUE**

Indicates the colourspace of the coded attribute values according to the ISO/IEC 23001-8 Codec Independent Code Points for ColourMatrix. When used in conjunction with ‘convertPlyColourspace=1’, a colourspace conversion will be performed at the input/output of the encoder and decoder if supported.

NB: the use of YCgCoR and ‘bitdepth=N’ implies that the bitdepth of the chroma component bitdepth is N + 1.

Value	RGB converter	Description
0	n/a	Direct coding (eg, RGB, XYZ)
1	Yes	YCbCr ITU-R BT.709
2	n/a	Unspecified
3	n/a	Reserved
4	No	USA Title 47 CFR 73.682 (a)(20)
5	No	YCbCr ITU-R BT.601
6	No	YCbCr SMPTE 170M
7	No	YCbCr SMPTE 240M
8	Yes (YCgCoR)	YCgCo / YCgCoR
9	No	YCbCr ITU-R BT.2020
10	No	YCbCr ITU-R BT.2020 (constant luminance)
11	No	YDzDx SMPTE ST 2085

**--attrScale=INT-VALUE` and `--attrOffset=INT-VALUE**

Scale and offset used to interpret coded attribute values. The encoder derives the coded attribute value as  $(attr - offset)/scale$ . The encoder and decoder scale coded attributes for output as  $attrscale + offset$ . NB: these parameters are only supported for reflectance attributes.

**--transformType=0|1|2**

Coding method to use for the current attribute:

Value	Description
0	Region Adaptive Hierarchical Transform (RAHT)
1	Hierarchical neighbourhood prediction
2	Hierarchical neighbourhood prediction as lifting transform
3	Uncompressed (PCM)

**--rahtPredictionEnabled=0|1**

Controls the use of transform domain prediction of RAHT coefficients from spatially upsampling the DC values of neighbouring parent nodes in the transform tree.

**--rahtPredictionThreshold0=0--19**

Controls a per-block threshold used to enable or disable the transform domain prediction of RAHT coefficients. This threshold specifies the number of parent neighbour points that must be present.

**--rahtPredictionThreshold1=0--19**

Controls a per-block threshold used to enable or disable the transform domain prediction of RAHT coefficients. This threshold specifies the number of neighbour points that must be present.

**--rahtSubnodePredictionEnabled=0|1**

Controls the use of transform domain prediction of RAHT coefficients from the DC values of sub-nodes of neighbouring parent nodes in the transform tree.

**--rahtPredictionSearchRange=INT-VALUE**

Controls the search range for neighbours search in RAHT.

**--rahtPredictionWeights=INT-VALUE-LIST**

A list of five weights that are used in the derivation of transform domain prediction of RAHT coefficients when subnode prediction is enabled.

**--numberOfNearestNeighboursInPrediction=INT-VALUE**

Attribute's maximum number of nearest neighbours to be used for prediction.

**--adaptivePredictionThreshold=INT-VALUE**

Neighbouring attribute value difference that enables the use of direct predictor selection over the weighted average. If bitdepth is greater than 8, the threshold is scaled by  $2^{(bitDepth - 8)}$ .

Applies to 'transformType=0' only.

**--predWeightBlending=0|1**

When enabled, blends the distance derived weights of the three-neighbour predictor according to the relative distances between the neighbours.

Applies to 'transformType=0' only.

**--direct\_avg\_predictor\_disabled\_flag=0|1**

Controls the use of the neighbour average predictor when direct prediction is invoked.

**--interComponentPredictionEnabled=0|1**

Controls the use of an in-loop inter-component prediction of attribute residuals. When enabled, the secondary attribute residuals (e.g. red/blue) are predicted from the primary component (e.g. green).

Applies to 'transformType=0' and 'attribute=color' only.

**--lastComponentPredictionEnabled=0|1**

Controls the use of an in-loop inter-component prediction of attribute coefficients. When enabled, the coefficient of the last component (e.g. Cr) of the secondary attribute is predicted from the corresponding first component (e.g. Cb) according to a simple model.

Applies to 'transformType=0 or 2' and 'attribute=color' only.

**--intraLodSearchRange=INT-VALUE**

Buffer range searched for nearest neighbours within the same level of detail. The value -1 configures a full-range search.

**--interLodSearchRange=INT-VALUE**

Buffer range searched for nearest neighbours between different levels of detail. The value -1 configures a full-range search.

**--max\_num\_direct\_predictors=INT-VALUE**

Maximum number of nearest neighbour candidates used in direct attribute prediction.

**--lodDecimator=0|1|2**

Controls the level-of-detail generation method:

Value	Description
0	No decimation is performed
1	Decimation by periodic lodSubsamplingPeriod
1	Decimation by distance to lodSubsamplingPeriod centroid

**--intraLodPredictionSkipLayers=INT-VALUE**

The number of detail levels where intra prediction is disabled, starting from the finest detail level. Applies to 'transformType=0' only.

Value	Description
-1	Disabled in all detail levels
0	Enabled in all detail levels
n	Disabled in n finest detail levels

**--aps\_scalable\_enabled\_flag=0|1**

Enable spatially scalable attribute encoding. The option is only valid when 'transformType=2', 'position-QpMultiplierLog2=3', 'lodDecimator=0', and 'trisoupNodeSizeLog2=0'.

**--max\_neigh\_range=INT-VALUE**

Limits the distance between a point and the neighbours used for its prediction. The maximum distance is expressed in units of node diagonals and is scaled according to the current level of detail.

**--levelOfDetailCount=INT-VALUE**

Attribute's number of levels of detail.

**--dist2=INT-VALUE**

An initial squared distances used to generate successive levels of detail. When equal to zero, an initial value is automatically determined.

**--dist2PercentileEstimate=FLOAT-VALUE**

Percentile of per-point nearest neighbour distances used to estimate 'dist2'.

**--positionQuantizationScaleAdjustsDist2=0|1**

Adjusts 'dist2' according to 'sequenceScale'. This option simplifies the specification of the per-attribute 'dist2' parameter.

The squared distance threshold used for generating levels-of-detail in attribute coding is dependent on the point cloud density and is therefore affected by geometry quantization. When this parameter is enabled, 'dist2' values are scaled by 'sequenceScale' squared, thereby allowing 'dist2' to be specified as an intrinsic property of the source sequence.

**--lodSubsamplingPeriod=INT-VALUE|INT-VALUE-LIST**

A list of sampling periods used to generate successive levels of detail.

**--canonical\_point\_order\_flag=0|1**

Controls the order used for attribute coding. The canonical (geometry decoding order) is usable only with LoD attribute coding and 'levelOfDetailCount=0'.

Value	Description
0	Morton order
1	Decoded geometry (canonical) order

**--spherical\_coord\_flag=0|1**

Controls the conversion of point co-ordinates used in attribute coding from the Cartesian domain to a scaled spherical domain.

**--attrSphericalMaxLog2=INT-VALUE**

Override spherical co-ordinate normalization factor. This may be used to compensate any increased azimuth resolution when 'predGeomAzimuthQuantization=1'.

Value	Description
0	Automatic calculation
1	Override max value

Applies when 'angularEnabled=1' and 'predGeomAzimuthQuantization=1'.

**--enableMortonCodeScaling=0|1**

Controls the activation of morton code scaling.

**--lod\_neigh\_bias=INT-VALUE-LIST**

A set of three bias factors corresponding to the first, second and third geometry axes used to weight nearest neighbours during the LoD generation and weighting processes. The value '1,1,1' implies no bias.

**--qp=INT-VALUE**

Attribute's luma quantization parameter.

**--qpChromaOffset=INT-VALUE**

Attribute's chroma quantization quantization parameter relative to luma. Only applies when 'attribute=colour'.

**--aps\_slice\_qp\_deltas\_present\_flag=0|1**

Enables signalling of per-slice QP values.

**--qpLayerOffsetsLuma=INT-VALUE-LIST**

Attribute's per layer luma QP offsets. A layer corresponds to a level-of-detail or RAHT transform block.

**--qpLayerOffsetsChroma=INT-VALUE-LIST**

Attribute's per layer chroma QP offsets. A layer corresponds to a level-of-detail or RAHT transform block. Only applies when 'attribute=colour'.

**--quantNeighWeight=INT-VALUE-LIST**

Three factors used to derive quantization weights when 'transformType=1'. The quantization weights are determined by recursively distributing each coefficient's weight to each of its neighbours, i, scaled by  $\text{quantNeighWeight}[i] \div 256$ .

**--predictionWithDistributionEnabled=0|1**  
Controls the activation of prediction within levels of detail based on the distribution of predictors.

**--max\_points\_per\_sort\_log2\_plus1=INT-VALUE**  
Specifies the maximum number of points per sort used in attribute coding based on morton code in case the number of LoDs is equal to 1.

**--attributeInterPredictionEnabled=0|1**  
Controls the activation of inter prediction of attribute.

**--attrInterPredSearchRange=INT-VALUE**  
Specifies the search range for nearest neighbour search in inter prediction candidate for attribute coding. A value of -1 indicates that the full range is used for the search.

**--rahtEnableCodeLayer=0|1**  
Controls the activation of inter frame inter layer prediction for RAHT, and when enabled, obtains the number of inter prediction layers based on the value of rahtInterPredictionDepthMinus1 and a preset value.

**--rahtEnableIntraPredCodeLayer=0|1**  
Controls the activation of non-pred layer in RD cost calculation for RAHT.

**--rahtInterPredictionDepthMinus1=INT-VALUE**  
Specifies the max depth to apply the inter prediction for RAHT.

**--attrInterPredTranslationThresh=INT-VALUE**  
(Encoder only) Specifies the maximum translation threshold used to disable inter prediction for attributes.

**--rahtInterSendFilters=0|1**  
Controls the enabling of inter-frame filters for RAHT.

**--rahtInterSkipFilteringLayers=INT-VALUE**  
Specifies the number of initial layers in RAHT to be skipped for inter-frame filtering.

**--rahtInterEligibilityLowerTh=INT-VALUE**  
Specifies the inter eligibility lower threshold for RAHT.

**--rahtInterEligibilityUpperTh=INT-VALUE**  
Specifies the inter eligibility upper threshold for RAHT.

**--rahtHybridPredictionEnabled=0|1**  
Specifies whether hybrid prediction is enabled in RAHT.

**--rahtHybridPredictionEnabledMinLayerMinus1=INT-VALUE**  
Specifies the minimum layer enabling hybrid prediction.

**--rahtHybridPredictionEnabledLayersCount=INT-VALUE**  
Specifies the number of layers that enable hybrid prediction in RAHT.

**--QPShiftStep=INT-VALUE**  
(Encoder only) Specifies the QP shift step used to derive the QP shift for attribute coding in inter predicted frames.

**--attrInterIntraSliceRDO=0|1**  
(Encoder only) Specifies whether slice level RDO is performed to select inter/intra prediction of a slice.

**--cross\_attr\_prediction\_enabled\_flag=0|1**  
Specifies whether (when 1) or not (when 0) attribute values shall be coded using correlations across different types of attributes when point clouds have multiple attributes.

**--attrMultiEncodedOrder=0|1**  
(Encoder only) Specifies the encode order for multiple attribute coding. 0: coding color prior to reflectance, 1: coding reflectance prior to color.

**--cross\_attr\_prediction\_enabled\_this\_type=0|1**  
Specifies whether (when 1) or not (when 0) the cross-attribute prediction is enabled for coding the current attribute if cross\_attr\_prediction\_enabled\_flag is 1.

**--refAttrIdx=INT-VALUE**  
Specifies the index of attribute identified by its attrIdx that is used for decoding the current attribute. It shall range from 0 to num\_attributes - 1 when cross\_attr\_prediction\_enabled\_this\_type is 1.



**--alphaY=FLOAT-VALUE**

Specifies the factor for computation of lagrangian multiplier for Y chanel.

**--alphaCb=FLOAT-VALUE**

Specifies the factor for computation of lagrangian multiplier for Cb chanel.

**--alphaCr=FLOAT-VALUE**

Specifies the factor for computation of lagrangian multiplier for Cr chanel.

**--alphaZeroTriplet=FLOAT-VALUE**

Specifies the factor for computation of lagrangian multiplier for Zero triplet decision.

**--alphaRef1=DOUBLE-VALUE**

Specifies the factor for computation of lagrangian multiplier for Reflectance chanel.

**--alphaZeroTripletRef1=DOUBLE-VALUE**

Specifies the factor for computation of lagrangian multiplier for Zero triplet decision for Reflectance.

**--chromaFormat420=0|1**

Specifies whether (when 1) or not (when 0) the chroma downsampling is enabled.

### Attribute recolouring (encoder only)

The following options configure the recolouring module, used when resampling a point cloud, or if the geometry coding process invents new points.

**--recolourSearchRange=INT-VALUE**

Attribute space search range for optimal attribute transfer.

**--recolourNumNeighboursFwd=INT-VALUE**

Number of source points used at the neighborhood of a target point to create the forward points list.

**--recolourNumNeighboursBwd=INT-VALUE**

Number of target points used at the neighborhood of a source point to create the backward points list.

**--recolourUseDistWeightedAvgFwd=0|1**

Use distance-weighted average for forward list.

**--recolourUseDistWeightedAvgBwd=0|1**

Use distance-weighted average for backward list.

**--recolourSkipAvgIfIdenticalSourcePointPresentFwd=0|1**

Do not use forward points list if an identical source point exists.

**--recolourSkipAvgIfIdenticalSourcePointPresentBwd=0|1**

Do not use backward points list if an identical source point exists.

**--recolourDistOffsetFwd=REAL-VALUE**

Distance offset to avoid infinite weight when distance between a forward list point and the target is zero.

**--recolourDistOffsetBwd=REAL-VALUE**

Distance offset to avoid infinite weight when distance between a backward list point and target is zero.

**--recolourMaxGeometryDist2Fwd=REAL-VALUE**

Maximum allowed squared distance of a source point from target to get into the forward list.

**--recolourMaxGeometryDist2Bwd=REAL-VALUE**

Maximum allowed squared distance of a source point from target to get into the backward list.

**--recolourMaxAttributeDist2Fwd=REAL-VALUE**

Maximum allowed squared attribute value difference of a source point for inclusion in the forward list.

**--recolourMaxAttributeDist2Bwd=REAL-VALUE**

Maximum allowed squared attribute value difference of a source point for inclusion in the backward list.

### layer-group slicing

The following options configure the layer-group slicing module.

**--layerGroupEnabledFlag=0|1**

Controls the enabling of layer-group based slice.

**--numLayerGroupsMinus1=INT-VALUE**  
Specifies the number of layer-groups.

**--numLayersPerLayerGroup=INT-VALUE | INT-VALUE-LIST**  
Specifies the number of coding layers in every layer-group.

**--subgroupBboxSize\_Cubic=INT-VALUE**  
(Encoder only) Specifies unit size of a cube which is used to initialize a subgroup bounding box.

**--enableSliceSelector=0 | 1 | 2**  
Controls partial coding. The value 1 enables encoder side FGS (fine granularity slices) selection and the value of 2 enables decoder side FGS selection. The value of 0 disables the slice selector (full decoding).

**--roiEnabledFlag=0 | 1**  
Controls the enabling of ROI based slice selection after encoding.

**--roiSize=INT-VALUE-LIST**  
(Encoder only) Specifies a list of length of a bounding box of a region of interest (ROI) for partial decoding of spatial region. Enabled only when roiEnabledFlag is enabled.

**--bboxScaleForRoiPoint=FLOAT-VALUE-LIST**  
(Encoder only) Specifies a list of scale factor to indicate the origin position of a ROI bounding box. The position is calculated by multiplying the parameters (ranges [0 : 1]) to the size of the bounding box in the x, y, and z-axis, respectively. Enabled only when roiEnabledFlag is enabled.

**--numSkipLayerGroups=INT-VALUE**  
Specifies the number of skipped layer-group from the leaf. 0 represents decoding full depth.

**--rootLayerGroupContextReferenceFlag=0 | 1**  
Controls the activation of context reference to the root subgroup. When enabled, subgroups refers to the saved context state of the root subgroup. When disabled, all subgroups except for the root subgroup refers to the saved context state of the parent subgroup.

**--depth1stSubgroupSearch=0 | 1**  
(Encoder only) Controls the order of FGSs. When enabled, FGSs are generated by depth first order. When disabled, FGSs are generated by breadth first order.

**--attributeContextReferenceIDPresentFlag=0 | 1**  
Controls context reference structure of attribute FGSs. When enabled, the context state reference structure of attribute FGSs is independent to that of geometry FGSs. When disabled, the context state reference of an attribute FGS is identical to that of a geometry FGS whose layer-group index and subgroup index are identical.

**--qpCoefDependentUnits=INT-VALUE**  
Specifies the coefficient of qp offsets for each dependent attribute data units.

**--qpLayerCoefDependentUnits=INT-VALUE**  
Specifies the coefficient of qp offsets for 1st layer of dependent attribute data units.

**--weightAdjustmentEnabledFlag=0 | 1**  
Controls the use of subgroup weight adjustment for near-lossless attribute coding.

**--weightAdjustmentMethod=0 | 1**  
(Encoder only) Controls weight adjustment coefficient generation method. When the value equals to 0, least mean squares based method is used. When the value equals to 1, linear regression based method is used with the constraint of matching highest weights.

**--geomAttrInterleavingEnabled=0 | 1**  
Specifies whether geometry and attribute data unit are interleaved during encoding.