
ISO/IEC JTC 1/SC 29/WG 11**Coding of moving pictures and audio****Convenorship: UNI (Italy)**

Document type: Approved WG 11 document**Title:** G-PCC test model v11 user manual**Status:** Final**Date of document:** 2020-08-03**Source:** 3DG**Expected action:** None**No. of pages:** 16**Email of convenor:** leonardo@chiariglione.org**Committee URL:** mpeg.chiariglione.org

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC 1/SC 29/WG 11 N19517
Online – June/July 2020

Source: 3DG

Title: G-PCC test model v11 user manual

Abstract

This document is a user manual describing usage of reference software for the G-PCC project. It applies to version 11.0 of the software.

Contents

1	General Information	4
1.1	Bug reporting	4
2	Obtaining the software	4
2.1	Building	4
2.1.1	Linux	4
2.1.2	OSX	4
2.1.3	Windows	4
3	Using the codec	5
3.1	General options	5
3.1.1	--help	5
3.1.2	--config=FILE, -c	5
3.1.3	--mode=VALUE	5
3.2	I/O parameters	5
3.2.1	--firstFrameNum=INT-VALUE	5
3.2.2	--frameCount=INT-VALUE	5
3.2.3	--uncompressedDataPath=FILE	6
3.2.4	--compressedStreamPath=FILE	6
3.2.5	--reconstructedDataPath=FILE	6
3.2.6	--postRecolourPath=FILE	6
3.2.7	--preInvScalePath=FILE	6
3.2.8	--outputBinaryPly=0 1	6
3.2.9	---outputResolution=REAL-VALUE	6
3.2.10	--convertPlyColourspace=0 1	6
3.2.11	--hack.reflectanceScale=0 1	6
4	Decoder-specific options	6
4.0.1	--skipOctreeLayers=INT-VALUE	6
4.0.2	--decodeMaxPoints=INT-VALUE	7
5	Encoder-specific options	7
5.0.1	--srcResolution=REAL-VALUE	7
5.0.2	--positionQuantizationScale=REAL-FACTOR	7
5.0.3	--positionQuantizationScaleAdjustsDist2=0 1	7

5.0.4	--seq_bounding_box_xyz0=x,y,z	7
5.0.5	--seq_bounding_box_whd=w,h,d	7
5.0.6	--mergeDuplicatedPoints=0 1	7
5.0.7	--geometry_axis_order=INT-VALUE	7
5.0.8	--disableAttributeCoding=0 1	8
5.0.9	--sortInputByAzimuth=0 1	8
5.0.10	--partitionMethod=0 2 3 4	8
5.0.11	--partitionNumUniformGeom=INT-VALUE	8
5.0.12	--partitionOctreeDepth=INT-VALUE	8
5.0.13	--sliceMaxPoints=INT-VALUE	8
5.0.14	--sliceMinPoints=INT-VALUE	8
5.0.15	--tileSize=INT-VALUE	8
5.0.16	--enforceLevelLimits=0 1	8
5.0.17	--cabac_bypass_stream_enabled_flag=0 1	9
5.0.18	--entropyContinuationEnabled=0 1	9
5.1	Geometry coding	9
5.1.1	--geomTreeType=0 1	9
5.1.2	--positionQuantisationEnabled=0 1	9
5.1.3	--positionBaseQp=INT-VALUE	9
5.1.4	--positionQpMultiplierLog2=0 1 2 3	9
5.1.5	--positionIdcmQp=INT-VALUE	9
5.1.6	--positionSliceQpOffset=INT-VALUE	9
5.1.7	--positionQuantisationOctreeDepth=INT-VALUE	9
5.1.8	qtbEnabled=0 1	9
5.1.9	maxNumQtBtBeforeOt=INT-VALUE	10
5.1.10	minQtbtSizeLog2=INT-VALUE	10
5.1.11	--bitwiseOccupancyCoding=0 1	10
5.1.12	--neighbourAvailBoundaryLog2=INT-VALUE	10
5.1.13	--inferredDirectCodingMode=0 1 2 3	10
5.1.14	--jointTwoPointIdcm=0 1	10
5.1.15	--adjacentChildContextualization=0 1	10
5.1.16	--intra_pred_max_node_size_log2=INT-VALUE	10
5.1.17	--planarEnabled=0 1	10
5.1.18	--planarModeIdcmUse=0--127	10
5.1.19	--planarModeThreshold0=0--127	10
5.1.20	--planarModeThreshold1=0--127	10
5.1.21	--planarModeThreshold2=0--127	11
5.1.22	--angularEnabled=0 1	11
5.1.23	--lidarHeadPosition=x,y,z	11
5.1.24	--numLasers=0--255	11
5.1.25	--lasersTheta=FLOAT-VALUE-LIST	11
5.1.26	--lasersZ=FLOAT-VALUE-LIST	11
5.1.27	--lasersNumPhiPerTurn=INT-VALUE-LIST	11
5.1.28	--planarBufferDisabled=0 1	11
5.1.29	--numOctreeEntropyStreams=INT-VALUE	11
5.1.30	--trisoupNodeSizeLog2=INT-VALUE INT-VALUE-LIST	11
5.1.31	--trisoup_sampling_value=INT-VALUE	12
5.1.32	--predGeomSort=INT-VALUE	12
5.1.33	--predGeomAzimuthSortPrecision=INT-VALUE	12
5.1.34	--predGeomTreePtsMax=INT-VALUE	12
5.1.35	--positionBaseQpFreqLog2=INT-VALUE	12

5.1.36	--positionSliceQpFreqLog2=INT-VALUE	12
5.1.37	--positionAzimuthScaleLog2=INT-VALUE	12
5.1.38	--positionRadiusInvScaleLog2=INT-VALUE	12
5.1.39	--positionAzimuthSpeed=INT-VALUE	12
5.1.40	--pointCountMetadata=0 1	12
5.2	Attribute coding	13
5.2.1	--attribute=NAME	13
5.2.2	--defaultValue=INT-VALUE-LIST	13
5.2.3	--colourMatrix=INT-VALUE	13
5.2.4	--bitdepth=INT-VALUE	13
5.2.5	--transformType=0 1 2	13
5.2.6	--rahtPredictionEnabled=0 1	14
5.2.7	--rahtPredictionThreshold0=0--19	14
5.2.8	--rahtPredictionThreshold1=0--19	14
5.2.9	--numberOfNearestNeighboursInPrediction=INT-VALUE	14
5.2.10	--adaptivePredictionThreshold=INT-VALUE	14
5.2.11	--interComponentPredictionEnabled=0 1	14
5.2.12	--lastComponentPredictionEnabled=0 1	14
5.2.13	--intraLoDSearchRange=INT-VALUE	14
5.2.14	--interLoDSearchRange=INT-VALUE	14
5.2.15	--max_num_direct_predictors=INT-VALUE	14
5.2.16	--lodDecimation=0 1	14
5.2.17	--intraLodPredictionEnabled=0 1	15
5.2.18	--aps_scalable_enabled_flag=0 1	15
5.2.19	--max_neigh_range=INT-VALUE	15
5.2.20	--levelOfDetailCount=INT-VALUE	15
5.2.21	--dist2=INT-VALUE	15
5.2.22	--dist2PercentileEstimate=FLOAT-VALUE	15
5.2.23	--lodSubsamplingPeriod=INT-VALUE INT-VALUE-LIST	15
5.2.24	--canonical_point_order_flag=0 1	15
5.2.25	--spherical_coord_flag=0 1	15
5.2.26	--lod_neigh_bias=INT-VALUE-LIST	15
5.2.27	--qp=INT-VALUE	15
5.2.28	--qpChromaOffset=INT-VALUE	15
5.2.29	--aps_slice_qp_deltas_present_flag=0 1	15
5.2.30	--qpLayerOffsetsLuma=INT-VALUE-LIST	15
5.2.31	--qpLayerOffsetsChroma=INT-VALUE-LIST	16
5.3	Attribute recolouring (encoder only)	16
5.3.1	--recolourSearchRange=INT-VALUE	16
5.3.2	--recolourNumNeighboursFwd=INT-VALUE	16
5.3.3	--recolourNumNeighboursBwd=INT-VALUE	16
5.3.4	--recolourUseDistWeightedAvgFwd=0 1	16
5.3.5	--recolourUseDistWeightedAvgBwd=0 1	16
5.3.6	--recolourSkipAvgIfIdenticalSourcePointPresentFwd=0 1	16
5.3.7	--recolourSkipAvgIfIdenticalSourcePointPresentBwd=0 1	16
5.3.8	--recolourDistOffsetFwd=REAL-VALUE	16
5.3.9	--recolourDistOffsetBwd=REAL-VALUE	16
5.3.10	--recolourMaxGeometryDist2Fwd=REAL-VALUE	16
5.3.11	--recolourMaxGeometryDist2Bwd=REAL-VALUE	16
5.3.12	--recolourMaxAttributeDist2Fwd=REAL-VALUE	16
5.3.13	--recolourMaxAttributeDist2Bwd=REAL-VALUE	16

General Information

Reference software is being made available to provide a reference implementation of the G-PCC standard being developed by MPEG (ISO/IEC SC29 WG11). 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 3DGC email reflector at mpeg-3dgc@gti.ssr.upm.es (registration required).

Bug reporting

Bugs should be reported on the issue tracker set up at <http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc13/issues>.

Obtaining the software

The authoritative location of the software is the following git repository: <http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc13>

Each released version may be identified by a version control system tag in the form `release-v${version}`.

An example:

```
$ git clone http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc13.git
$ cd mpeg-pcc-tmc13
$ git checkout release-v4.0
```

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.

---outputResolution=REAL-VALUE

The output point clouds resolution in points per metre. Decoded point clouds that indicate a coded resolution are rescaled to this output resolution.

--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.

--hack.reflectanceScale=0|1

Some input data uses 8-bit reflectance data scaled by 255 and represented using 16-bit attributes. This option enables a conversion of 16-bit to 8-bit at the encoder, and the corresponding conversion from 8-bit back to 16-bit at the decoder. If the original data has been scaled by 255, the conversion process is lossless.

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`).

Encoder-specific options

--srcResolution=REAL-VALUE

Resolution of the input point cloud in points per metre. This value is used to derive the resolution of the coded point cloud.

If equal to zero, the input point cloud size is considered to be dimensionless. In this case, the coded point cloud will indicate the dimensionless scale factor of the coded size to the input size.

--positionQuantizationScale=REAL-FACTOR

Prior to encoding, scale the point cloud geometry by multiplying each co-ordinate by the real *FACTOR* and rounding to integer precision. The scale factor is written to the bitstream and a decoder may use it to provide output at the original scale.

NB: when using trisoup geometry coding, use `triSoupIntToOrigScale` instead of this option.

--positionQuantizationScaleAdjustsDist2=0|1

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 `positionQuantizationScale` squared, thereby allowing `dist2` to be specified as an intrinsic property of the source sequence.

--seq_bounding_box_xyz0=x,y,z

Explicitly sets the origin of the sequence-level bounding box in unscaled integer coordinates.

NB: This option has no effect if `seq_bounding_box_whd=0,0,0`.

--seq_bounding_box_whd=w,h,d

Explicitly sets the size of the sequence-level bounding box in unscaled integer coordinates.

When w, h, d not equal to `0,0,0`, the sequence-level bounding box origin is set according to `seq_bounding_box_xyz0`. Otherwise, the sequence-level bounding box is determined by the encoder.

--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.

--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

Value	Coding order
6	y, x, z
7	x, y, z

--disableAttributeCoding=0|1

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

--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.

--partitionMethod=0|2|3|4

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

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.

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.

--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.

--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.

Geometry coding

`--geomTreeType=0|1`

Selects the geometry tree coding method.

Value	Description
0	Octree
1	Predictive geometry tree

`--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`.

`--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.

`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 of 0 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--127

Controls the frequency of IDCM eligibility. A value of zero prevents IDCM being eligible for use, while the value 127 does not further constrain IDCM use.

--planarModeThreshold0=0--127

Controls the eligibility threshold of the first planar mode based upon local child node density.

--planarModeThreshold1=0--127

Controls the eligibility threshold of the second planar mode based upon local child node density.

--planarModeThreshold2=0--127

Controls the eligibility threshold of the third planar mode based upon local child node density.

--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 of 0 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`.

--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

--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 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`.

--pointCountMetadata=0|1

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

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 $2^{**}(\text{bitdepth}-1)$.

--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.

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

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

--bitdepth=INT-VALUE

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

--transformType=0|1|2

Coding method to use for the current attribute:

Value	Description
0	Hierarchical neighbourhood prediction
1	Region Adaptive Hierarchical Transform (RAHT)

Value	Description
2	Hierarchical neighbourhood prediction as lifting transform

`--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.

`--numberOfNearestNeighboursInPrediction=INT-VALUE`

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

`--adaptivePredictionThreshold=INT-VALUE`

Neighbouring attribute value difference that enables choice of single|multi predictors. Applies to transformType=2 only. A value of -1 is replaced by $2^{*(\text{bitdepth}-2)}$.

`--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=2 and attribute=color only.

`--intraLoDSearchRange=INT-VALUE`

Buffer range searched for nearest neighbours within the same level of detail.

`--interLoDSearchRange=INT-VALUE`

Buffer range searched for nearest neighbours between different levels of detail.

`--max_num_direct_predictors=INT-VALUE`

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

`--lodDecimation=0|1`

Controls the level-of-detail generation method:

Value	Description
0	Euclidean distance thresholding (dist2)
1	Decimation by lodSubsamplingPeriod

--intraLodPredictionEnabled=0|1

Part of LoD attribute coding. Permits (1) points to be predicted from previously reconstructed points within the same LoD.

--aps_scalable_enabled_flag=0|1

Enable spatially scalable attribute encoding. The option is only valid when `transformType=2`, `lodDecimation=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`.

--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.

--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 is 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.

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.