
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 v7 user manual

Status: Final

Date of document: 2019-09-05

Source: 3DG

Expected action: None

No. of pages: 9

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 N18664
Gothenburg, SE – July 2019

Source: 3DG

Title: G-PCC test model v7 user manual

Abstract

This document is a user manual describing usage of reference software for the G-PCC project. It applies to version 7.0 of the software.

Contents

1	General Information	2
1.1	Bug reporting	3
2	Obtaining the software	3
2.1	Building	3
2.1.1	Linux	3
2.1.2	OSX	3
2.1.3	Windows	3
3	Using the codec	4
3.1	General options	4
3.1.1	--help	4
3.1.2	--config=FILE, -c	4
3.1.3	--mode=VALUE	4
3.2	I/O parameters	4
3.2.1	--uncompressedDataPath=FILE	4
3.2.2	--compressedStreamPath=FILE	4
3.2.3	--reconstructedDataPath=FILE	4
3.2.4	--postRecolourPath=FILE	4
3.2.5	--preInvScalePath=FILE	4
3.2.6	--outputBinaryPly=0 1	5
3.2.7	--colourTransform=0 1	5
3.2.8	--hack.reflectanceScale=0 1	5
4	Decoder-specific options	5
5	Encoder-specific options	5
5.0.1	--positionQuantizationScale=REAL-FACTOR	5
5.0.2	--positionQuantizationScaleAdjustsDist2=0 1	5
5.0.3	--seq_bounding_box_xyz0=x,y,z	5
5.0.4	--seq_bounding_box_whd=w,h,d	5
5.0.5	--mergeDuplicatedPoints=0 1	5
5.0.6	--disableAttributeCoding=0 1	6
5.0.7	--partitionMethod=0 2 3	6
5.0.8	--partitionNumUniformGeom=INT-VALUE	6

5.0.9	--partitionOctreeDepth=INT-VALUE	6
5.0.10	--sliceMaxPoints=INT-VALUE	6
5.0.11	--sliceMinPoints=INT-VALUE	6
5.0.12	--tileSize=INT-VALUE	6
5.0.13	--cabac_bypass_stream_enabled_flag=0 1	6
5.1	Geometry coding	6
5.1.1	--bitwiseOccupancyCoding=0 1	6
5.1.2	--neighbourContextRestriction=0 1	6
5.1.3	--neighbourAvailBoundaryLog2=INT-VALUE	7
5.1.4	--inferredDirectCodingMode=0 1	7
5.1.5	--adjacentChildContextualization=0 1	7
5.1.6	--intra_pred_max_node_size_log2=INT-VALUE	7
5.1.7	--ctxOccupancyReductionFactor=INT-VALUE	7
5.1.8	--trisoup_node_size_log2=INT-VALUE	7
5.2	Attribute coding	7
5.2.1	--attribute=NAME	7
5.2.2	--bitdepth=INT-VALUE	7
5.2.3	--transformType=0 1 2	7
5.2.4	--rahtDepth=INT-VALUE	8
5.2.5	--rahtPredictionEnabled=0 1	8
5.2.6	--numberOfNearestNeighboursInPrediction=INT-VALUE	8
5.2.7	--adaptivePredictionThreshold=INT-VALUE	8
5.2.8	--attributeSearchRange=INT-VALUE	8
5.2.9	--max_num_direct_predictors=INT-VALUE	8
5.2.10	--lodDecimation=0 1	8
5.2.11	--intraLodPredictionEnabled=0 1	8
5.2.12	--levelOfDetailCount=INT-VALUE	8
5.2.13	--dist2=INT-VALUE INT-VALUE-LIST	8
5.2.14	--qp=INT-VALUE	8
5.2.15	--qpChromaOffset=INT-VALUE	8
5.2.16	--aps_slice_qp_deltas_present_flag=0 1	8
5.2.17	--qpLayerOffsetsLuma=INT-VALUE-LIST	9
5.2.18	--qpLayerOffsetsChroma=INT-VALUE-LIST	9
5.3	Attribute recolouring (encoder only)	9
5.3.1	--recolourSearchRange=INT-VALUE	9
5.3.2	--recolourNumNeighboursFwd=INT-VALUE	9
5.3.3	--recolourNumNeighboursBwd=INT-VALUE	9
5.3.4	--recolourUseDistWeightedAvgFwd=0 1	9
5.3.5	--recolourUseDistWeightedAvgBwd=0 1	9
5.3.6	--recolourSkipAvgIfIdenticalSourcePointPresentFwd=0 1	9
5.3.7	--recolourSkipAvgIfIdenticalSourcePointPresentBwd=0 1	9
5.3.8	--recolourDistOffsetFwd=REAL-VALUE	9
5.3.9	--recolourDistOffsetBwd=REAL-VALUE	9
5.3.10	--recolourMaxGeometryDist2Fwd=REAL-VALUE	9
5.3.11	--recolourMaxGeometryDist2Bwd=REAL-VALUE	9
5.3.12	--recolourMaxAttributeDist2Fwd=REAL-VALUE	9
5.3.13	--recolourMaxAttributeDist2Bwd=REAL-VALUE	9

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 a PLY file describing a point cloud 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 a 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

--uncompressedDataPath=FILE

(Encoder only) The input source point cloud to be compressed.

--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.

--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.

--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.

--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.

--colourTransform=0|1

Controls the use of a colour space transformation before attribute coding and after decoding.

Value	Description
0	none
1	RGB to YCbCr (Rec.709)

--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

There are no decoder specific options at this time.

Encoder-specific options

--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.

--disableAttributeCoding=0|1

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

--partitionMethod=0|2|3

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

--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.

--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

Geometry coding

--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.

--neighbourContextRestriction=0|1

Octree occupancy coding is contextualised in part by the occupancy of neighbouring octree nodes. The neighbour context restriction limits the use of neighbouring nodes to direct octree siblings.

NB: This option conflicts with `neighbourAvailBoundaryLog2`. It is necessary to set `neighbourAvailBoundaryLog2=0` when `neighbourContextRestriction=1`.

--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 indicates that no constraint is applied.

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

--inferredDirectCodingMode=0|1

Controls the use of early termination of the geometry octree for isolated points.

--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.

--ctxOccupancyReductionFactor=INT-VALUE

Adjusts the number of contexts used in bit-wise occupancy coding. The total number of contexts used is 256 » *VALUE*.

NB: the final standard is expected to define this factor as a constant.

--trisoup_node_size_log2=INT-VALUE

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.

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.

--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)
2	Hierarchical neighbourhood prediction as lifting transform

--rahtDepth=INT-VALUE

Number of bits for Morton representation of RAHT co-ordinate components.

--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.

--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 transform-Type=2 only. A value of -1 is replaced by $2^{*(\text{bitdepth}-2)}$.

--attributeSearchRange=INT-VALUE

Range for nearest neighbour search.

--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
1	Decimation by 1:3

--intraLodPredictionEnabled=0|1

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

--levelOfDetailCount=INT-VALUE

Attribute's number of levels of detail.

--dist2=INT-VALUE|INT-VALUE-LIST

Attribute's list of squared distances, or initial value for automatic derivation.

--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 is 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.