

**ISO/IEC JTC 1/SC 29/WG 11**

**Coding of moving pictures and audio**

**Convenorship: UNI (Italy)**

**ISO/IEC JTC 1/SC 29/WG 11 w19371**

**Document type: Approved WG 11 document**

**Title:**

**Status: Approved**

**Date of document: 2020-06-01**

**Source: 3DG**

**Expected action:**

**No. of pages: 17**

**Email of convenor: leonardo@chiariglione.org**

**Committee URL: mpeg.chiariglione.org**

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 w19371**

**April 2020, Alpbach**

|  |  |
| --- | --- |
| **Source** | **3DG** |
| **Title** | WD of V-PCC Reference Software |

**WD of V-PCC Reference Software**

# Summary

This document is a user manual describing usage of reference software for the V-PCC project. It applies to version 10.0 of the software.

# General Information

Reference software is being made available to provide a reference implementation of the V3C 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 V3C 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).

Bugs should be reported on the issue tracker set up at <http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2/issues>.

# Obtaining the software

## Clone

The authoritative location of the software is the following git repository: <http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2>

Each released version may be identified by a version control system tag in the form release-v10.0 [1].

An example:

git clone http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-tmc2.git

cd mpeg-pcc-tmc2

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.

### Scripts

Bash scripts can be use to build mpeg-pcc-dmetric project: build.sh to build solutions and clear.sh to clean.

### Linux

mkdir build cd build cmake .. make

../bin/PccAppEncoder --help

../bin/PccAppDecoder --help

../bin/PccAppMetrics –help

### OSX

mkdir build cd build

cmake .. -G Xcode xcodebuild

../bin/PccAppEncoder --help

../bin/PccAppDecoder --help

../bin/PccAppMetrics –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.

### HM reference software

The common test conditions use HM reference software to encode the created videos. To respect the CTC, we must use the HM: HM-16.20+SCM-8.8 and apply a patch to this version to activate the 3D motion estimation. The patch can be found in the subfolder: mpeg-pcc-tmc2/dependencies/hm-modification/pcc\_me-ext\_for\_HM-16.20+SCM-8.8.patch.

The next command lines could be used to download HM reference software and apply patch:

svn checkout https://hevc.hhi.fraunhofer.de/svn/svn\_HEVCSoftware/\\ tags/HM-16.20+SCM-8.8/external/HM-16.20+SCM-8.8+3DMC;

cd external/HM-16.20+SCM-8.8+3DMC

svn patch ../../mpeg-pcc-tmc2/dependencies/hm-modification/\\ pcc\_me-ext\_for\_HM-16.20+SCM-8.8.patch

### HDRTools

The HDRTools is used to perform color conversion. This software must be cloned and built, and a path must be set to the PccAppEndoder and PccAppDecoder to perform CTC conditions [2].

git clone -b 0.17-dev https://gitlab.com/standards/HDRTools.git

### Metrics

To evaluate the performance of the VPCC encoding, the PCC\_distorsion software must be used:

git clone http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-dmetric.git

# Structure of the test model

This software consists of multiple executables and libraries. The two main applications are PccAppEncoder and PccAppDecoder, that are the applications used to encode and decode the point clouds. These softwares are directly linked to the two main libraries: PccLibEncoder and PccLibDecoder that contains the main encoder and decoder processes. The processes and the data structures used both by encoder and decoder are in the PccLibCommon library. The VPCC bitstreams data structures are defined in library: PccLibBitstreamCommon. The encoder and decoder bitstream writing and reading processes are stored in to separate libraries: PccLibBitstreamWriter and PccLibBitstreamReader.

Additional of these two libraries, the virtual wrapper libraries are used to encode video (PccLibVideoEncoder), to decode video (PccLibVideoDecoder) and to perform color conversion of the video streams (PccLibColorConverter). These libraries are wrappers and are used to launch external applications or libraries: HDRTools, HM Encoder/Decoder (TLibEncoder/TLibDecoder). These libraries defined application program interfaces (API) to easily launch external processes to perform these tasks without change the VPCC source code.

The below figure is the module dependency graph of the VPCC reference software.

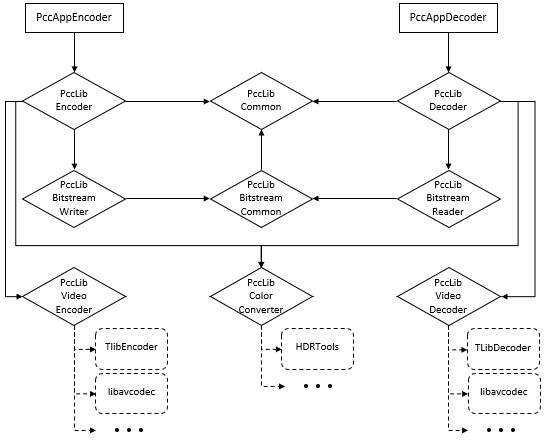


Figure 1 – Structure of the test model

# Using the codec

../bin/PccAppEncoder [--help] [-c config.cfg] [--parameter=value]

../bin/PccAppDecoder [--help] [--parameter=value]

../bin/PccAppMetrics [--help] [--parameter=value]

## Principle

The encoder takes as input a PLY file describing a point cloud with integer positions and, optionally, per-point integer color attributes.

The output of the encoder is a binary bitstream encapsulated using the V3C annex-B format.

Conversely, the decoder takes as input a compressed bitstream file in V3C 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.

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.

## Common test condition configurations

The configuration files stored in the ./cfg/ sub-folder could be used to perform the V3C common test conditions (CTC) experiments. An example of the usage of this file could be found in ./test/runme\_linux.sh.

The reference software configuration did not define the CTC, please validated your experiments based on the W19324 - V3C common test conditions document [1].

## Examples

### Encoder

The next command line encodes one streams:

./bin/PccAppEncoder \

--config=./cfg/common/ctc-common.cfg \

--config=./cfg/condition/ctc-all-intra.cfg \

--config=./cfg/sequence/queen.cfg \

--config=./cfg/rate/ctc-r1.cfg \

--colorTransform=0 \

--configurationFolder=./cfg/ \

--uncompressedDataFolder=./People/ \

--colorSpaceConversionPath=HDRConvert \

--videoEncoderPath=TAppEncoderHighBitDepthStatic \

--videoEncoderOccupancyMapPath=TAppEncoderHighBitDepthStatic \

--compressedStreamPath=./S22C2AI\_queen/S22C2AIR01\_queen.bin \ --frameCount=32

To compute the metrics in the encode, the normal of the source point cloud must be given to the encoder. The next parameter must be added to the previous command:

--normalDataPath=./People/Technicolor/queen\_n/frame\_%04d\_n.ply

### Decoder

The next command line decodes one streams:

./bin/PccAppDecoder \

--startFrameNumber=0000 \

--compressedStreamPath=./S22C2AI\_queen/S22C2AIR01\_queen.bin \

--reconstructedDataPath=./S22C2AI\_queen/S22C2AIR01\_queen\_dec\_%04d.ply \

--videoDecoderPath=TAppDecoderHighBitDepthStatic \

--videoDecoderOccupancyMapPath=TAppDecoderHighBitDepthStatic \

--colorSpaceConversionPath=./external/HDRTools/bin/HDRConvert \

--inverseColorSpaceConversionConfig=./cfg/hdrconvert/yuv420torgb444.cfg \

--nbThread=1 \

--colorTransform=0 \

To compute the metrics in the decoder, the normal of the source point cloud and the source PLY must be given to the decoder. The next parameter must be added to the previous command:

--config=./cfg/sequence/queen.cfg \

--uncompressedDataFolder=./People/ \

--normalDataPath=./People/Technicolor/queen\_n/frame\_%04d\_n.ply

### Metrics

PccAppMetrics could be used to test the PccLibMetrics. For CTC experiments, it’s sugested to used mpeg-pcc-dmetrics: http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-dmetric.git.

For example, mpeg-pcc-dmetric and PccAppMetric could be used with the next command line:

../bin/PccAppMetrics \

--uncompressedDataPath=longdress\_vox10\_1051.ply \

--reconstructedDataPath=./S26C2AIR01\_longdress\_dec\_1051.ply \

--normalDataPath=./People/8i/longdress\_n/longdress\_vox10\_1051\_n.ply \ --resolution=1023 \

--frameCount=1

./mpeg-pcc-demetric/test/pc\_error \

--fileA=./People/8i/8iVFBv2/longdress/Ply/longdress\_vox10\_1051.ply \

--fileB=S26C2AIR01\_longdress\_dec\_1051.ply \

--inputNorm=./People/8i/longdress\_n/longdress\_vox10\_1051\_n.ply \

--color=1 \

--resolution= 1023

The two softwares give the same results.

### Scripts

More examples of running could be found in ./test/runme\_linux.sh.

These examples can be start based on your system with the following scripts:

* ./test/runme\_linux.sh
* ./test/runme\_windows.bat
* ./test/runme\_osx.sh

The V3C common test condition (CTC) command lines could be found in ./test/ctc\_command\_line.sh.

## General options

The next tables show the parameters of the encoder, decoder, and metrics programs.

### Encoder parameters

|  |  |
| --- | --- |
| **Parameter=Value** | **Usage** |
| help | This help text |
| **Global** |  |
| c|config | Configuration file name |
| configurationFolder | Folder where the configuration files are stored,use for cfg relative paths. |
| uncompressedDataFolder | Folder where the uncompress input data are stored, use for cfg relative paths. |
| uncompressedDataPath | Input pointcloud to encode. Multi-frame sequences may be represented by %04i |
| compressedStreamPath | Output(encoder)/Input(decoder) compressed bitstream |
| reconstructedDataPath | Output decoded pointcloud. Multi-frame sequences may be represented by %04i |
| startFrameNumber | First frame number in sequence to encode/decode |
| frameCount | Number of frames to encode |
| groupOfFramesSize | Random access period |
| colorTransform | The colour transform to be applied: 0: none  1: RGB to YCbCr (Rec.709) |
| colorSpaceConversionPath | Path to the HDRConvert. If unset, an internal color space conversion is used |
| colorSpaceConversionConfig | HDRConvert configuration file used for RGB444 to YUV420 conversion |
| inverseColorSpaceConversion | HDRConvert configuration file used for YUV420 |
| Config | to RGB444 conversion |
| videoEncoderPath | HM video encoder executable |
| videoEncoderAuxPath | HM video encoder executable |
| videoEncoderOccupancyMapPath | HM lossless video encoder executable for occupancy map |

|  |  |
| --- | --- |
| nbThread | Number of thread used for parallel processing |
| keepIntermediateFiles | Keep intermediate files: RGB, YUV and bin |
| **Encoder** |  |
| nnNormalEstimation | Number of points used for normal estimation |
| gridBasedRefineSegmentation | Use grid-based approach for segmentation refinement |
| maxNNCountRefineSegmentation | Number of nearest neighbors used during segmentation refinement |
| iterationCountRefine | Number of iterations performed during |
| Segmentation | segmentation refinement |
| voxelDimensionRefine | Voxel dimension for segmentation refinement |
| Segmentation | (must be a power of 2) |
| searchRadiusRefineSegmentation | Search radius for segmentation refinement |
| occupancyResolution | Resolution of packing block(a block contain only one patch) |
| enablePatchSplitting | Enable patch splitting |
| maxPatchSize | Maximum patch size for segmentation |
| log2QuantizerSizeX | log2 of Quantization step for patch size X:  0. pixel precision 4.16 as before |
| log2QuantizerSizeY | log2 of Quantization step for patch size Y:  0. pixel precision 4.16 as before |
| minPointCountPerCCPatch | Minimum number of points for a connected |
| Segmentation | component to be retained as a patch |
| maxNNCountPatchSegmentation | Number of nearest neighbors used during connected components extraction |
| surfaceThickness | Surface thickness |
| depthQuantizationStep | minimum level for patches |
| maxAllowedDist2RawPoints | Maximum distance for a point to be ignored |
| Detection | during raw points detection |
| maxAllowedDist2RawPoints | Maximum distance for a point to be ignored |
| Selection | during raw points selection |
| lambdaRefineSegmentation | Controls the smoothness of the patch boundaries during segmentation refinement |
| minimumImageWidth | Minimum width of packed patch frame |
| minimumImageHeight | Minimum height of packed patch frame |
| maxCandidateCount | Maximum nuber of candidates in list L |
| occupancyPrecision | Occupancy map B0 precision |
| occupancyMapVideoEncoderConfig | Occupancy map encoder config file |
| occupancyMapQP | QP for compression of occupancy map video |
| enhancedOccupancyMapCode | Use enhanced-delta-depth code |
| EOMFixBitCount | enhanced occupancy map fixed bit count |
| occupancyMapRefinement | Use occupancy map refinement |
| postprocessSmoothingFilterType | Exclude geometry smoothing from attribute transfer |
| flagGeometrySmoothing | Enable geometry smoothing |
| neighborCountSmoothing | Neighbor count smoothing |
| radius2Smoothing | Radius to smoothing |
| radius2BoundaryDetection | Radius to boundary detection |
| thresholdSmoothing | Threshold smoothing |
| patchExpansion | Use occupancy map refinement |
| gridSmoothing | Enable grid smoothing |
| gridSize | grid size for the smoothing |
| thresholdColorSmoothing | Threshold of color smoothing |
| cgridSize | grid size for the color smoothing |
| thresholdColorDifference | Threshold of color difference between cells |
| thresholdColorVariation | Threshold of color variation in cells |
| flagColorSmoothing | Enable color smoothing |
| thresholdColorPreSmoothing | Threshold of color pre-smoothing |
| thresholdColorPreSmoothing | Threshold of color pre-smoothing local |
| LocalEntropy | entropy |
| radius2ColorPreSmoothing | Redius of color pre-smoothing |
| neighborCountColorPreSmoothing | Neighbor count for color pre-smoothing |
| flagColorPreSmoothing | Enable color pre-smoothing |
| bestColorSearchRange | Best color search range |
| numNeighborsColorTransferFwd | Number of neighbors creating Fwd list |
| numNeighborsColorTransferBwd | Number of neighbors creating Bwd list |
| useDistWeightedAverageFwd | Distance weighted average for Fwd list |
| useDistWeightedAverageBwd | Distance weighted average for Bwd list |
| skipAvgIfIdenticalSourcePoint | Skip avgeraging if target is identical to a |
| PresentFwd | Fwd point |
| skipAvgIfIdenticalSourcePoint | Skip avgeraging if target is identical to a |
| PresentBwd | Bwd point |
| distOffsetFwd | Distance offset to avoid infinite weight |
| distOffsetBwd | Distance offset to avoid infinite weight |
| maxGeometryDist2Fwd | Maximum allowed distance for a Fwd point |
| maxGeometryDist2Bwd | Maximum allowed distance for a Bwd point |
| maxColorDist2Fwd | Maximum allowed pari-wise color distance for  Fwd list |
| maxColorDist2Bwd | Maximum allowed pari-wise color distance for  Bwd list |
| excludeColorOutlier | Exclude color outliers from the NN set |
| thresholdColorOutlierDist | Threshold of color distance to exclude outliers from the NN set |
| geometryQP | QP for compression of geometry video |
| textureQP | QP for compression of texture video |
| geometryConfig | HM configuration file for geometry compression |
| geometryD0Config | HM configuration file for geometry D0 compression |
| geometryD1Config | HM configuration file for geometry D1 compression |
| textureConfig | HM configuration file for texture compression |
| textureT0Config | HM configuration file for texture D0 compression |
| textureT1Config | HM configuration file for texture D1 compression |
| losslessGeo | Enable lossless encoding of geometry |
| noAttributes | Disable encoding of attributes |
| losslessGeo444 | Use 4444 format for lossless geometry |
| useRawPointsSeparateVideo | Compress raw points with video codec |
| textureRawSeparateVideoWidth | Width of the MP’s texture in separate video |

geometryMPConfig HM configuration file for raw points geometry compression

|  |  |
| --- | --- |
| textureMPConfig | HM configuration file for raw points texture compression |
| absoluteD1 | Absolute D1 |
| absoluteT1 | Absolute T1 |
| multipleStreams | number of video(geometry and attribute) streams |
| qpT1 | qp adjustment for T1 0, +3, -3... |
| qpD1 | qp adjustment for D1 : 0, +3, -3... |
| constrainedPack | Temporally consistent patch packing |
| levelOfDetailX | levelOfDetail : X axis in 2D space (should be greater than 1) |
| levelOfDetailY | levelOfDetail : Y axis in 2D space (should be greater than 1) |
| groupDilation | Group Dilation |
| textureDilationOffLossless | Group Dilation |
| offsetLossyOM | Value to be assigned to non-zero occupancy map positions |
| thresholdLossyOM | Threshold for converting non-binary occupancy map to binary |
| prefilterLossyOM | Selects whether the occupany map is prefiltered before lossy compression (default=false) |
| patchColorSubsampling | Enable per patch color sub-sampling |
| deltaCoding | Delta meta-data coding |
| maxNumRefAtalsList | maximum Number of Reference Patch list, default: 1 |
| maxNumRefAtlasFrame | maximum Number of Reference Atlas Frame per list, default: 1 |
| pointLocalReconstruction | Use point local reconstruction |
| mapCountMinus1 | Numbers of layers (rename to maps?) |
| singleLayerPixelInterleaving | Use single layer pixel interleaving |
| removeDuplicatePoints | Remove duplicate points( |
| surfaceSeparation | surface separation |
| highGradientSeparation | Separate high gradient points from a patch |
| minGradient | Minimun gradient for a point to be separated |
| minNumHighGradientPoints | Minimum number of connected high gradient points to be separated from a patch |
| packingStrategy | Patches packing strategy(0: anchor packing, 1(default): flexible packing, 2: tetris packing) |
| useEightOrientations | Allow either 2 orientations (0(default): NULL  AND SWAP), or 8 orientation (1) |
| safeGuardDistance | Number of empty blocks that must exist between the patches (default=1) |
| textureBGFill | Selects the background filling operation for texture only (0: patch-edge extension, 1(default): smoothed push-pull algorithm), 2:  harmonic background filling |
| lossyRawPointsPatch | Lossy raw points patch(0: no lossy raw points patch, 1: enable lossy raw points patch  (default=0) |
| minNormSumOfInvDist4MP  Selection | Minimum normalized sum of inverse distance for raw points selection: double value between 0.0 and 1.0 (default=0.35) |
| lossyRawPointPatchGeoQP | QP value for geometry in lossy raw points patch (default=4) |
| globalPatchAllocation | Global temporally consistent patch allocation.(0: anchor’s packing method(default), 1: gpa algorithm, 2: gtp algorithm) |
| globalPackingStrategyGOF | Number of frames to pack globally (0:(entire  GOF)) |
| globalPackingStrategyReset | Remove the reference to the previous frame  (0(default), 1) |
| globalPackingStrategyThreshold | matched patches area ratio threshold (decides if connections are valid or not, 0(default)) |
| patchPrecedenceOrder | Order of patches |
| lowDelayEncoding | Low Delay encoding (0(default): do nothing, 1: does not allow overlap of patches bounding boxes for low delay encoding) |
| geometryPadding | Selects the background filling operation for geometry (0: anchor, 1(default): 3D geometry padding) |
| apply3dMotionCompensation | Use auxilliary information for 3d motion compensation.(0: conventional video coding,  1: 3D motion compensated) |
| geometry3dCoordinatesBitdepth | Bit depth of geomtery 3D coordinates |
| geometryNominal2dBitdepth | Bit depth of geometry 2D |
| nbPlrmMode | Number of PLR mode |
| patchSize | Size of Patch for PLR |
| enhancedProjectionPlaneUse | Enhanced Projection Plane(0: OFF, 1: ON) |
| minWeightEPP | Minimum value |
| additionalProjectionPlaneMode | additiona Projection Plane Mode 0:none 1:Y-Axis 2:X-Axis 3:Z-Axis 4:All-Axis 5:apply to portion |
| partialAdditionalProjectionPlane | The value determines the partial point cloud. It’s available with only additionalProjectionPlaneMode(5) |
| enablePointCloudPartitioning |  |
| roiBoundingBoxMinX |  |
| roiBoundingBoxMaxX |  |
| roiBoundingBoxMinY |  |
| roiBoundingBoxMaxY |  |
| roiBoundingBoxMinZ |  |
| roiBoundingBoxMaxZ |  |
| numTilesHor |  |
| tileHeightToWidthRatio |  |
| numCutsAlong1stLongestAxis |  |
| numCutsAlong2ndLongestAxis |  |
| numCutsAlong3rdLongestAxis |  |
| mortonOrderSortRawPoints |  |
| pbfEnableFlag | enable patch block filtering |
| pbfFilterSize | pbfFilterSize |
| pbfPassesCount | pbfPassesCount |
| pbfLog2Threshold | pbfLog2Threshold |
| **Metrics** |  |
| computeChecksum | Compute checksum |
| computeMetrics | Compute metrics |
| normalDataPath | Input pointcloud to encode. Multi-frame sequences may be represented by %04i |
| resolution | Specify the intrinsic resolution |
| dropdups | 0(detect), 1(drop), 2(average) subsequent points with same coordinates |
| neighborsProc | 0(undefined), 1(average), 2(weighted average), 3(min), 4(max) neighbors with same geometric distance |
| **4.4.2 Decoder parameters** |  |
| **Parameter=Value** | **Usage** |
| help | This help text |
| **Global** |  |
| c,config | Configuration file name |
| compressedStreamPath | Input  compressed bitstream |
| reconstructedDataPath | Output decoded pointcloud. Multi-frame sequences may be represented by %04i |
| startFrameNumber | Fist frame number in sequence to encode/decode |
| colorTransform | The colour transform to be applied:  0: none  1: RGB to YCbCr (Rec.709) |
| colorSpaceConversion | Path to the HDRConvert. If unset, an |
| Path | internal color space conversion is used |
| inverseColorSpaceConversion | HDRConvert configuration file used for |
| Config | YUV420 to RGB444 conversion |
| videoDecoderPath= | HM video decoder executable |
| videoDecoderOccupancyMap | HM lossless video decoder executable |
| Path | for occupancy map |
| nbThread | Number of thread used for parallel processing |
| keepIntermediateFiles | Keep intermediate files: RGB, YUV and bin |
| testLevelOfDetail | Disable patch sampling resolution |
| Signaling | scaling; use in conjunction with same parameter in encoder |
| patchColorSubsampling | Enable per-patch color up-sampling |
| **Metrics** |  |
| computeChecksum=1 | Compute checksum |
| computeMetrics=1 | Compute metrics |

uncompressedDataFolder Folder where the uncompress input data are stored, use for cfg relative

paths.

|  |  |
| --- | --- |
| startFrameNumber | Fist frame number in sequence to encode/decode |
| frameCount | Number of frames to encode |
| groupOfFramesSize | Random access period |
| uncompressedDataPath | Input pointcloud to encode. Multi-frame sequences may be represented by %04i |
| reconstructedDataPath | Output decoded pointcloud. Multi-frame sequences may be represented by %04i |
| normalDataPath | Input pointcloud to encode. Multi-frame sequences may be represented by %04i |
| resolution | Specify the intrinsic resolution |
| dropdups | 0(detect), 1(drop), 2(average) subsequent points with same coordinates |
| neighborsProc | 0(undefined), 1(average), 2(weighted average), 3(min), 4(max) neighbors with same geometric distance |
| nbThread | Number of thread used for parallel processing |
| minimumImageHeight | Ignore parameter |
| flagColorPreSmoothing | Ignore parameter |
| surfaceSeparation | Ignore parameter |
| **4.4.3 Metrics parameters** |  |
| **Parameter=Value** | **Usage** |
| help | This help text |
| computeChecksum | Compute checksum |
| computeMetrics | Compute metrics |
| startFrameNumber | Fist frame number in sequence to |
|  | encode/decode |
| frameCount | Number of frames to encode |
| uncompressedDataPath | Input pointcloud to encode. Multi-frame |
|  | sequences may be represented by %04i |
| reconstructedDataPath | Output decoded pointcloud. Multi-frame |
|  | sequences may be represented by %04i |
| normalDataPath | Input pointcloud to encode. Multi-frame |
|  | sequences may be represented by %04i |
| resolution | Specify the intrinsic resolution |
| dropdups | 0(detect), 1(drop), 2(average) subsequent |
|  | points with same coordinates |
| neighborsProc | 0(undefined), 1(average), 2(weighted |
|  | average), 3(min), 4(max) neighbors with |
|  | same geometric distance |
| nbThread | Number of thread used for parallel |
|  | processing |
| minimumImageHeight | Ignore parameter |
| flagColorPreSmoothing | Ignore parameter |
| surfaceSeparation | Ignore parameter |

# References

[1] “Common Test Conditions for PCC”, ISO/IEC JTC1/SC29/WG11 MPEG/W19324, Apr. 2020, Alpbach.

[2] “V-PCC Test Model v10”, ISO/IEC JTC1/SC29/WG11 MPEG/W19325, Apr. 2020, Alpbach.