**Moving Picture Experts Group (MPEG)**
**of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11** Document: MPEG-w18666
July 2019, Gothenburg, Sweden

| | | |
|---|---|---|
| *Title:* | V-PCC test model v7.0 user manual | |
| *Status:* | Draft | |
| *Purpose:* | User manual | |
| *Author(s):* | Julien Ricard | *julien.ricard@interdigital.com* |
| *Source:* | 3DG | |

## Abstract

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

## Contents



## General Information

Reference software is being made available to provide a reference implementation of the V-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 V-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-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-v7.0`.

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/exter
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.p
```

# 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 colour attributes.

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

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

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 to perform the V-PCC 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 not defined the CTC, please validated your experimentes based on the wxxxxxx- V-PCC common test conditions document.

## 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.

## General options

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

**Encoder parameters**

| Parameter=Value | Usage |
|---|---|
| –help=0 | 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 compressed bitstream |
| –reconstructedDataPath=“” | Output decoded pointcloud. Multi-frame sequences may be represented by %04i |
| –startFrameNumber=0 | First frame number in sequence to encode/decode |
| –frameCount=300 | Number of frames to encode |
| –groupOfFramesSize=32 | Random access period |
| –colorTransform=1 | The colour transform to be applied:<br>0: none<br>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 |
| –inverseColorSpaceConversionConfig=“” | HDRConvert configuration file used for YUV420 to RGB444 conversion |
| –videoEncoderPath=“” | HM video encoder executable |
| –videoEncoderOccupancyMapPath=“” | HM lossless video encoder executable for occupancy map |
| –nbThread=1 | Number of thread used for parallel processing |
| –keepIntermediateFiles=0 | Keep intermediate files: RGB, YUV and bin |
| **Encoder** | |
| –nnNormalEstimation=16 | Number of points used for normal estimation |
| –maxNNCountRefineSegmentation=256 | Number of nearest neighbors used during segmentation refinement |
| –iterationCountRefineSegmentation=100 | Number of iterations performed during segmentation refinement |
| –occupancyResolution=16 | Resolution T of the occupancy map |
| –minPointCountPerCCPatch Segmentation=16 | Minimum number of points for a connected component to be retained as a patch |
| –maxNNCountPatchSegmentation=16 | Number of nearest neighbors used |

| | |
|---|---|
| | during connected components extraction |
| –surfaceThickness=4 | Surface thickness |
| –maxAllowedDepth=255 | Maximum depth per patch |
| –maxAllowedDist2MissedPointsDetection=9 | Maximum distance for a point to be ignored during missed point detection |
| –maxAllowedDist2MissedPointsSelection=1 | Maximum distance for a point to be ignored during missed points selection |
| –lambdaRefineSegmentation=3 | Controls the smoothness of the patch boundaries during segmentation refinement |
| –minimumImageWidth=1280 | Minimum width of packed patch frame |
| –minimumImageHeight=1280 | Minimum height of packed patch frame |
| –maxCandidateCount=4 | Maximum nuber of candidates in list L |
| –occupancyPrecision=4 | Occupancy map B0 precision |
| –occupancyMapVideoEncoderConfig=“” | Occupancy map encoder config file |
| –occupancyMapQP=8 | QP for compression of occupancy map video |
| –useOccupancyMapVideo=1 | compress occupancy map with video codec |
| –neighborCountSmoothing=64 | todo(kmammou) |
| –radius2Smoothing=64 | todo(kmammou) |
| –radius2BoundaryDetection=64 | todo(kmammou) |
| –thresholdSmoothing=64 | todo(kmammou) |
| –gridSmoothing=1 | Enable grid smoothing |
| –thresholdColorSmoothing=10 | Threshold of color smoothing |
| –thresholdLocalEntropy=4.5 | Threshold of local entropy |
| –radius2ColorSmoothing=64 | Redius of color smoothing |
| –neighborCountColorSmoothing=64 | Neighbor count for color smoothing |
| –flagColorSmoothing=0 | Enable color smoothing |
| –thresholdColorPreSmoothing=10 | Threshold of color pre-smoothing |
| –thresholdColorPreSmoothing LocalEntropy=4.5 | Threshold of color pre-smoothing local entropy |
| –radius2ColorPreSmoothing=64 | Redius of color pre-smoothing |
| –neighborCountColorPreSmoothing=64 | Neighbor count for color pre-smoothing |
| –flagColorPreSmoothing=1 | Enable color pre-smoothing |
| –bestColorSearchRange=0 | todo(kmammou) |
| –geometryQP=28 | QP for compression of geometry video |
| –textureQP=43 | 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 |
| –losslessGeo=0 | Enable lossless encoding of geometry |
| –losslessTexture=0 | Enable lossless encoding of texture |
| –noAttributes=0 | Disable encoding of attributes |

| | |
|---|---|
| –losslessGeo444=0 | Use 4444 format for lossless geometry |
| –useMissedPointsSeparateVideo=0 | compress missed point with video codec |
| –geometryMPConfig=“” | HM configuration file for missed points geometry compression |
| –textureMPConfig=“” | HM configuration file for missed points texture compression |
| –absoluteD1=1 | Absolute D1 |
| –constrainedPack=1 | Temporally consistent patch packing |
| –binArithCoding=1 | Binary arithmetic coding |
| –testLevelOfDetail=0 | Force non-zero level of detail for testing |
| –testLevelOfDetailSignaling=0 | Test the patch resolution signaling with pseudo-random values |
| –groupDilation=1 | Group Dilation |
| –textureDilationOffLossless=1 | Group Dilation |
| –enhancedDeltaDepthCode=0 | Use enhanced-delta-depth code |
| –patchColorSubsampling=0 | Enable per patch color sub-sampling |
| –deltaCoding=1 | Delta meta-data coding |
| –projectionMode=0 | projectionMode 0:min, 1:max, 2:adaptive frame and patch, 3:adaptive patch (all frames) |
| –oneLayerMode=0 | Use one layer mode |
| –singleLayerPixelInterleaving=0 | Use single layer pixel interleaving |
| –removeDuplicatePoints=1 | Remove duplicate points |
| –sixDirectionMode=1 | Use Six Direction Projection mode |
| –surfaceSeparation=0 | surface separation |
| –packingStrategy=1 | Patches packing strategy (0: anchor packing, 1(default): flexible packing, 2: tetris packing) |
| –useEightOrientations=0 | Allow either 2 orientations (0(default): NULL AND SWAP), or 8 orientation (1) |
| –safeGuardDistance=0 | Number of empty blocks that must exist between the patches (default=1) |
| –textureBGFill=1 | Selects the background filling operation for texture only (0: patch-edge extension, 1(default): smoothed push-pull algorithm), 2: harmonic background filling |
| –lossyMissedPointsPatch=0 | Lossy missed points patch(0: no lossy missed points patch, 1: enable lossy missed points patch (default=0) |
| –minNormSumOfInvDist4MP Selection=0.35 | Minimum normalized sum of inverse distance for missed points selection: double value between 0.0 and 1.0 (default=0.35) |
| –lossyMppGeoQP=4 | QP value for geometry in lossy missed points patch (default=4) |
| –globalPatchAllocation=0 | Global temporally consistent patch allocation. (0: anchor’s packing method(default), 1: gpa algorithm) |
| –apply3dMotionCompensation=1 | Use auxilliary information for 3d motion compensation.(0: conventional video coding, |

| | |
|---|---|
| | 1: 3D motion compensated) |
| –geometry3dCoordinatesBitdepth=10 | Bit depth of geomtery 3D coordinates |
| –geometryNominal2dBitdepth=8 | Bit depth of geometry 2D |
| –nbPlrmMode=0 | Number of PLR mode |
| –patchSize=0 | Size of Patch for PLR |
| –enhancedProjectionPlane=1 | Use enhanced Projection Plane<br>(0: OFF, 1: ON) |
| –minWeightEPP=0.6 | Minimum value |
| –additionalProjectionPlaneMode=0 | additiona Projection Plane Mode 0:none<br>1:Y-Axis 2:X-Axis 3:Z-Axis 4:All-Axis<br>5:apply to portion |
| –partialAdditionalProjectionPlane=0 | The value determines the partial point cloud. It's available with only additionalProjectionPlaneMode(5) |
| **Metrics** | |
| –computeChecksum=1 | Compute checksum |
| –computeMetrics=1 | Compute metrics |
| –normalDataPath="" | Input pointcloud to encode.<br>Multi-frame sequences may be represented by %04i |
| –resolution=1023 | Specify the intrinsic resolution |
| –dropdups=2 | 0(detect), 1(drop), 2(average) subsequent points with same coordinates |
| –neighborsProc=1 | 0(undefined), 1(average), 2(weighted average), 3(min), 4(max) neighbors with same geometric distance |

**Decoder parameters**

| **Parameter=Value** | **Usage** |
|---|---|
| –help=0 | This help text |
| **Global** | |
| -c,–config=... | Configuration file name |
| –compressedStreamPath="" | Input<br>compressed bitstream |
| –reconstructedDataPath="" | Output decoded pointcloud. Multi-frame sequences may be represented by %04i |
| –startFrameNumber=0 | Fist frame number in sequence to encode/decode |
| –colorTransform=1 | The colour transform to be applied:<br>0: none<br>1: RGB to YCbCr (Rec.709) |
| –colorSpaceConversionPath="" | Path to the HDRConvert. If unset, an internal color space conversion is used |
| –inverseColorSpaceConversionConfig="" | HDRConvert configuration file used for YUV420 to RGB444 conversion |
| –videoDecoderPath="" | HM video decoder executable |
| –videoDecoderOccupancyMapPath="" | HM lossless video decoder executable for occupancy map |
| –nbThread=1 | Number of thread used for parallel |

| | |
|---|---|
| | processing |
| –keepIntermediateFiles=0 | Keep intermediate files: RGB, YUV and bin |
| **Metrics** | |
| –testLevelOfDetailSignaling=0 | Disable patch sampling resolution scaling; use in conjunction with same parameter in encoder |
| –patchColorSubsampling=0 | 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=0 | Fist frame number in sequence to encode/decode |
| –frameCount=0 | Number of frames to encode |
| –groupOfFramesSize=32 | 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=1023 | Specify the intrinsic resolution |
| –dropdups=2 | 0(detect), 1(drop), 2(average) subsequent points with same coordinates |
| –neighborsProc=1 | 0(undefined), 1(average), 2(weighted average), 3(min), 4(max) neighbors with same geometric distance |
| –nbThread=0 | Number of thread used for parallel processing |
| –minimumImageHeight=0 | Ignore parameter |
| –flagColorPreSmoothing=0 | Ignore parameter |
| –surfaceSeparation=0 | Ignore parameter |

**Metrics parameters**

| **Parameter=Value** | **Usage** |
|---|---|
| –help=0 | This help text |
| –computeChecksum=1 | Compute checksum |
| –computeMetrics=1 | Compute metrics |
| –startFrameNumber=0 | Fist frame number in sequence to |
| | encode/decode |
| –frameCount=0 | 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=1023 | Specify the intrinsic resolution |
| –dropdups=2 | 0(detect), 1(drop), 2(average) subsequent |
| | points with same coordinates |
| –neighborsProc=1 | 0(undefined), 1(average), 2(weighted |
| | average), 3(min), 4(max) neighbors with |
| | same geometric distance |
| –nbThread=0 | Number of thread used for parallel |
| | processing |
| –minimumImageHeight=0 | Ignore parameter |
| –flagColorPreSmoothing=0 | Ignore parameter |
| –surfaceSeparation=0 | Ignore parameter |