INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 7 MPEG 3D GRAPHICS CODING

ISO/IEC JTC 1/SC 29/WG 7 N01312
October 2025, Geneva

Title:	Video-based dynamic mesh coding test model v14.0 user manual	
Source:	WG 7, MPEG 3D Graphics Coding	
Purpose:	User manual	
Status:	Draft	
Author(s):	Wenjie Zou	*wjzou@xidian.edu.cn*
	Jungsun Kim	*jungsun_kim@apple.com*

Video-based dynamic mesh coding test model v14.0 user manual

Abstract

This document is a user manual describing usage of reference software for the V-DMC project. It applies to version 14.0 of the software.

Contents

1 General Information **3**

2 Obtaining the software **3**

3 Building **3**
- 3.1 Building script . . . 3
- 3.2 Build manually . . . 4
  - 3.2.1 OSX . . . 4
  - 3.2.2 Linux . . . 4
  - 3.2.3 Windows . . . 4
- 3.3 Dependencies . . . 4

4 Architecture **5**
- 4.1 Core libraries . . . 5
- 4.2 Wrapper libraries . . . 6
- 4.3 Applications . . . 6
  - 4.3.1 Main applications . . . 6
  - 4.3.2 Wrapper applications . . . 6
  - 4.3.3 Unit test applications . . . 7

5 Usage **7**
- 5.1 Encode . . . 7
- 5.2 Decode . . . 8
- 5.3 Runtime configuration and configuration files . . . 8
- 5.4 Run experiment . . . 8
- 5.5 Collect results . . . 13
- 5.6 Run all experiments and create render and graph pdf files . . . 14
- 5.7 View decoded sequences . . . 15

6 Main software input parameters **17**
- 6.1 Encode software input parameters . . . 17
- 6.2 Decode software input parameters . . . 27
- 6.3 Metrics software input parameters . . . 28

7 Other informations **29**
- 7.1 Licence . . . 29
- 7.2 Documentation . . . 30
- 7.3 Issue reporting . . . 30

1 General Information

Reference software is being made available to provide a reference implementation of the video-based dynamic mesh coding standard being developed by MPEG-3DG (ISO/IEC SC29 WG7).

One of the main goals of the reference software is to provide a basis upon which to conduct experiments in order to determine which coding tools provide desired coding performance. It is not meant to be a particularly efficient implementation of anything, and one may notice its apparent unsuitability for a particular use. It should not be construed to be a reflection of how complex a production-quality implementation of a future 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 3DG email reflector at `mpeg-3dgc@gti.ssr.upm.es` (registration required).

2 Obtaining the software

The authoritative location of the software is the following git repository: `https://github.com/MPEGGroup/mpeg-vdmc-tm`

Each released version may be identified by a version control system tag in the form: `v14.0`

An example:

```
$ git clone \
   https://github.com/MPEGGroup/mpeg-vdmc-tm.git
$ cd mpeg-vmesh-tm
$ git checkout v14.0
```

> 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.

3 Building

3.1 Building script

A bash script is provided to facilitate the building operations.

To build V-DMC test model softwares with this script please use the following command line:

```
$ ./build.sh
$ ./build.sh --help
./build.sh mpeg-vmesh-tm building script:

    Usage:
       -h|--help     : Display this information.
       -o|--ouptut   : Output build directory.
       -n|--ninja    : Use Ninja.
       --debug       : Build in debug mode.
       --release     : Build in release mode.
       --doc         : Build documentation (latex and pdflatex requiered).
       --format      : Format source code.
       --tidy        : Check source code with clang-tidy.
       --cppcheck    : Check source code with cppcheck.
       --test        : Build unit tests.
       --meshType=*  : Define template mesh type: float or double.
```

```
      --codeCodecId: Code codec id used in the bitstream.

    Examples:
      ../build.sh
      ../build.sh --debug
      ../build.sh --doc
      ../build.sh --format
```

Another script could be used to clean the current solutions with the following command lines:

```
$ ./clear.sh      # Remove ./build/ sub-folder.
$ ./clear.sh all  # Remove all cloned dependencies.
```

3.2 Build manually

Standard CMake build commands can be used to build the software depending on the system you used.

3.2.1 OSX

```
$ mkdir build
$ cmake -S. -Bbuild -G Xcode
$ xcodebuild -project build/vmesh.xcodeproj -configuration Debug
```

3.2.2 Linux

```
$ mkdir build
$ cmake -DCMAKE_BUILD_TYPE=Release -S. -Bbuild/Release
$ cmake --build ./build/Release --config Release --parallel 12
```

3.2.3 Windows

```
$ md build
$ cmake -DCMAKE_BUILD_TYPE=Release -S. -Bbuild/Release
$ cmake --build ./build/Release --config Release --parallel 12
```

3.3 Dependencies

The V-DMC test model software uses several dependencies that are cloned and patched by the CMake building process.

These dependencies are:

URL	Commit/tag
DirectX-Headers	1b79ddaeabc4b16c772ca63adc5bdf7d5f741460
DirectXMath	b404898c9dcaff7b686bbaf6d2fba8ff0184a17e
DirectXMesh	2c0ed18e271afa99a70948f784dfe082127fa0de
UVAtlas	5af1b5d2a0fd9e0e5d17aa0971ab17c890e318e0
mpeg-pcc-mmetric	1_1_7
tinyply	2.3.4
HDRTools	v0.23
HM	HM-16.21+SCM-8.8
VVCSoftware_VTM	VTM-21.2
vvenc	v1.7.0
draco	1af95a20b81624f64c4b19794cb3ca991e6d0a76
vvdec	v1.6.0
SHM	SHM-12.4

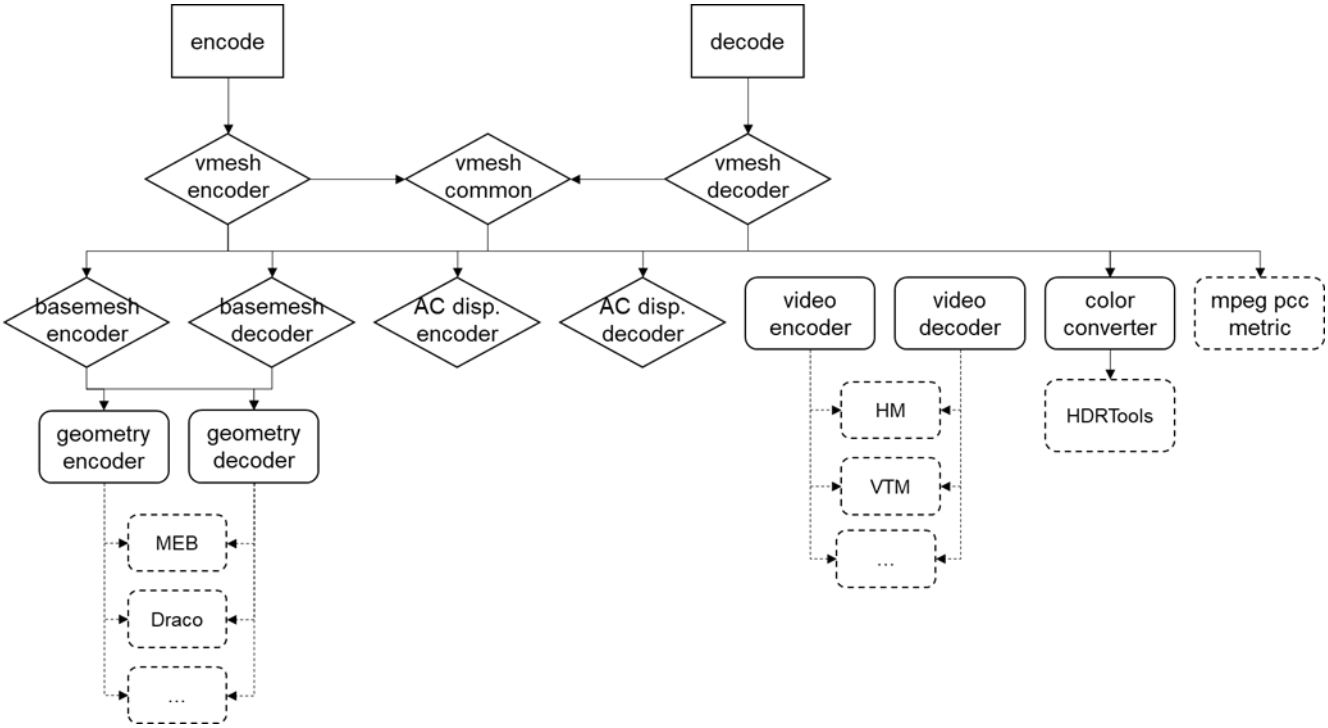

Figure 1 – Scheme of architecture of V-Mesh test model.

4 Architecture

V-Mesh test model software is organized as shows in figure 1.

4.1 Core libraries

The core codec processes are grouped into twenty one libraries:

- `bitstreamsCommon` containing the fundamental bitstream parsing and generation processes.
- `v3cBitstream` containing the bitstream parsing and generation processes for V3C.
- `vmeshCommon` containing the util objects and the processes shared by V-Mesh encoding and decoding processes.
- `vmeshEncoder` containing the V-Mesh encoding processes.
- `vmeshDecoder` containing the V-Mesh decoding processes.
- `vmeshConformance` containing the conformance verification processes.
- `acDisplacementBitstream` containing the bitstream parsing and generation processes for arithmetic coding of displacement.
- `acDisplacementCommon` containing the common processes for arithmetic coding of displacement.
- `acDisplacementEncoder` containing the arithmetic encoding processes of displacement.
- `acDisplacementDecoder` containing the arithmetic decoding processes of displacement.
- `atlasBitstream` containing the bitstream parsing and generation processes for atlas information.
- `atlasCommon` containing the common processes for atlas information.
- `atlasEncoder` containing the atlas information encoding processes.
- `atlasDecoder` containing the atlas information decoding processes.
- `basemeshBitstream` containing the bitstream parsing and generation processes for basemesh.
- `basemeshEncoder` containing the basemesh encoding processes.
- `basemeshDecoder` containing the basemesh decoding processes.
- `ebBitstream` containing the bitstream parsing and generation processes for Edge Breaker.
- `ebCommon` containing the processes shared by MPEG Edge Breaker (MEB) encoding and decod-

ing processes.
- `ebEncoder` containing the MEB encoding processes.
- `ebDecoder` containing the MEB decoding processes.

4.2 Wrapper libraries

To unify interfaces with external libraries used to encode/decode meshes, encode/decode/convert videos and compute metrics, wrapper libraries have been created:

- `videoEncoder`: wrapper to HM encoder, VTM encoder;
- `videoDecoder`: wrapper to HM decoder, VTM decoder;
- `geometryEncoder`: wrapper to MEB encoder;
- `geometryDecoder`: wrapper to MEB decoder;
- `colourConverter`: wrapper to HDRTools.

These libraries are based on a virtual object that can be derived to implement one specific interface with the external libraries. The source codes of the wrapper libraries are stored in the `source/wrapper/` sub-folder.

4.3 Applications

The source codes of V-Mesh applications are stored in the `source/app/` sub-folder.

4.3.1 Main applications

The two main application of the V-Mesh test model are:

- `encode`: that can be used to encode mesh sequence to a V-Mesh bitstream.
- `decode`: that decode V-Mesh bitstream.
- `metric`: that can be used to evaluates V-mesh compression performance metrics.

The following section shows examples of the usage of these softwares.

4.3.2 Wrapper applications

To evaluate the wrapper libraries specific applications have been created. These applications can be used to crosscheck the usage of the external applications. The source code of these applications are in `source/app/wrapper/` sub-folder:

- `ebEncode`;
- `ebDecode`;
- `videoEncoder`;
- `videoDecoder`;
- `colourConverte`;
- `objToPlyConvert`;
- `dracoEncoder`;
- `dracoDecoder`;
- `checksum`;
- `baseMeshEncodeApp`;
- `baseMeshDecodeApp`;
- `atlasEncodeApp`;
- `atlasDecodeApp`;
- `acdEncodeApp`;
- `acdDecodeApp`.

4.3.3 Unit test applications

To evaluate the source code and to guarantee an early regression detection, unit test application has been created.

The unit test application is based on Google Testing Framework. The source code of this software is stored in `source/app/unitTests/`. The list of the unit tests that are implemented can be logged with the following command line:

```
$ ./build/Release/bin/unitTests --gtest_list_tests
draco.
  encode
  decode
metrics.
  compare
hm.
  disp
  disp2
  texture
colourConvert.
  hdrToolsUp
  hdrToolsDown
vmesh.
  all
```

The unit tests can be executed with:

```
$ ./build/Release/bin/unitTests -v 0
...
[       OK ] hm.texture (4770 ms)
[----------] 3 tests from hm (5595 ms total)

[----------] 2 tests from colourConvert
[ RUN      ] colourConvert.hdrToolsUp
[       OK ] colourConvert.hdrToolsUp (190 ms)
[ RUN      ] colourConvert.hdrToolsDown
[       OK ] colourConvert.hdrToolsDown (197 ms)
[----------] 2 tests from colourConvert (388 ms total)

[----------] 1 test from vmesh
[ RUN      ] vmesh.all
[       OK ] vmesh.all (49463 ms)
[----------] 1 test from vmesh (49463 ms total)

[----------] Global test environment tear-down
[==========] 9 tests from 5 test suites ran. (64516 ms total)
[  PASSED  ] 9 tests.
```

Note: it's greatly recommended to execute the unit test application before each submission in the repository.

5 Usage

5.1 Encode

The encode command line is the following one:

```
$ ./build/Release/bin/encode \
    --config=./generatedConfigFiles/s3c1r3_bask/encoder.cfg \
    --frameCount=1 \
    --compressed=s3c1r3_bask.vmesh
```

5.2 Decode

The decode can be executed with:

```
./build/Release/bin/decode \
  --config=./generatedConfigFiles/s3c1r3_bask/decoder.cfg \
  --compressed=s3c1r3_bask.vmesh \
  --decMesh=s3c1r3_bask_%04d_dec.obj \
  --decTex=s3c1r3_bask_%04d_dec.png \
  --decMat=s3c1r3_bask_%04d_dec.mtl \
```

5.3 Runtime configuration and configuration files

To generate the configuration files (conmon test conditions) according to your system paths, the following action must be made:

1. copy and edit `cfg/cfg-site-default.yaml` as `cfg/cfg-site.yaml`, the paths for the binaries, sequence prefix, and the external tool configuration prefix;
2. run the ./scripts/gen-cfg.sh' script:

```
$ ./scripts/gen-cfg.sh \
    --cfgdir=./cfg/ \
    --outdir=/path/to/generated/cfgfiles
```

This operation can be executed with script './scripts/create_configuration_files.sh' and in this case the file 'cfg/cfg-site.yaml' is generated automatically according to the current folder.

```
$ ./scripts/create_configuration_files.sh
. /scripts/create_configuration_files.sh Create configuration files:

  Usage:
    -o|--outdir=: configured directory      (default: config/ )
    -s|--seqdir=: source sequence directory (default:  )
    -c|--codec=:  video codec: hm, vtm      (default: hm )

  Examples:
    ./scripts/create_configuration_files.sh
    ./scripts/create_configuration_files.sh \
      --outdir=generatedConfigFilesHM  \
      --seqdir=/path/to/contents/voxelized/ \
      --codec=hm
    ./scripts/create_configuration_files.sh \
      --outdir=generatedConfigFilesVTM \
      --seqdir=/path/to/contents/voxelized/ \
      --codec=vtm
```

5.4 Run experiment

An example script (`scripts/run.sh`) demonstrates how to launch the entire toolchain for a single job in the configured experiment.

This scripts starts:

- encoding process
- decoding process
- pcc metrics computation
- ibsm metrics computation

The usage of this script are presented below:

```
$ ./scripts/run.sh

    Usage:
       -h|--help   : print help
       -q|--quiet  : disable logs            (default: 1 )
       -f|--frames : frame count             (default: 1 )
       -c|--cfgdir : configured directory    (default: "" )
       -o|--outdir : output directory        (default: "results" )
       --condId=   : condition: 1, 2         (default: 1 )
       --seqId=    : seq: 1,2,3,4,5,6,7,8    (default: 1 )
       --rateId=   : Rate: 1,2,3,4,5         (default: 1 )
       --tmmMetric : Use TMM metric software (default: 0 )
       --render    : Create rendered images  (default: 0 )
       --encParams : configured directory    (default: "" )
       --decParams : configured directory    (default: "" )
       --csv       : generate .csv file      (default: "" )

    Examples:
      - ../scripts/run.sh
      - ../scripts/run.sh \
          --condId=1 \
          --seqId=3 \
          --rateId=3 \
          --cfgdir=generatedConfigFiles
      - ../scripts/run.sh \
          --condId=1 \
          --seqId=3 \
          --rateId=3 \
          --cfgdir=generatedConfigFiles \
          --TMMMETRIC
```

Note: The preceding script uses the `mpeg-pcc-mmetric` software and this dependency can be cloned and built with the following command line:

```
$ ./scripts/get_external_tools.sh
```

A example of execution of this script is:

```
$ ./scripts/run.sh \
    --condId=1 \
    --seqId=3 \
    --rateId=3 \
    --cfgdir=generatedConfigFiles \
    --outdir=results
Run vmesh encoder/decoder/metrics: ./scripts
Encode: results/F001/s3c1r2_bask/s3c1r2_bask
./build/Release/bin/encode \
    --config=./generatedConfigFiles/s3c1r2_bask//encoder.cfg \
```

```
    --frameCount=1 \
    --compressed=results/F001/s3c1r2_bask/s3c1r2_bask.vmesh \
   > results/F001/s3c1r2_bask/encoder.log 2>&1
Decode: results/F001/s3c1r2_bask/s3c1r2_bask
./build/Release/bin/decode \
    --config=./generatedConfigFiles/s3c1r2_bask//decoder.cfg \
    --compressed=results/F001/s3c1r2_bask/s3c1r2_bask.vmesh \
    --decMesh=results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.obj \
    --decTex=results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.png \
    --decMat=results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.mtl \
   > results/F001/s3c1r2_bask/decoder.log 2>&1
Metrics IBSM: results/F001/s3c1r2_bask/s3c1r2_bask
./externaltools/mpeg-pcc-mmetric/build/mm \
   sequence \
    --firstFrame    1 \
    --lastFrame     1 \
   END \
   dequantize \
    --inputModel    /path/to/contents/basketball_player_fr%04d_qp12_qt12.obj \
    --outputModel   ID:deqRef \
    --useFixedPoint \
    --qp            12 \
    --minPos        "-725.812988 -483.908997 -586.02002" \
    --maxPos        "1252.02002 1411.98999 1025.34998" \
    --qt            12 \
    --minUv         "0 0" \
    --maxUv         "1.0 1.0" \
   END \
   dequantize \
    --inputModel    results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.obj \
    --outputModel   ID:deqDis \
    --useFixedPoint \
    --qp            12 \
    --minPos        "-725.812988 -483.908997 -586.02002" \
    --maxPos        "1252.02002 1411.98999 1025.34998" \
    --qt            12 \
    --minUv         "0 0" \
    --maxUv         "1.0 1.0" \
   END \
   compare \
    --mode          ibsm \
    --inputModelA   ID:deqRef \
    --inputModelB   ID:deqDis \
    --inputMapA     /path/to/contents/basketball_player_fr%04d.png \
    --inputMapB     results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.png \
    --outputCsv     results/F001/s3c1r2_bask/metric_ibsm.csv \
   > results/F001/s3c1r2_bask/metric_ibsm.log
Metrics PCC: results/F001/s3c1r2_bask/s3c1r2_bask
./externaltools/mpeg-pcc-mmetric/build/mm \
   sequence \
    --firstFrame    1 \
    --lastFrame     1 \
   END \
```

```
dequantize \
 --inputModel    /path/to/contents/basketball_player_fr%04d_qp12_qt12.obj \
 --outputModel   ID:deqRef \
 --useFixedPoint \
 --qp            12 \
 --qt            12 \
 --minPos        "-725.812988 -483.908997 -586.02002" \
 --maxPos        "1252.02002 1411.98999 1025.34998" \
 --minUv         "0.0 0.0" \
 --maxUv         "1.0 1.0" \
END \
dequantize \
 --inputModel    results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.obj \
 --outputModel   ID:deqDis \
 --useFixedPoint \
 --qp            12 \
 --qt            12 \
 --minPos        "-725.812988 -483.908997 -586.02002" \
 --maxPos        "1252.02002 1411.98999 1025.34998" \
 --minUv         "0.0 0.0" \
 --maxUv         "1.0 1.0" \
END \
reindex \
 --inputModel    ID:deqRef \
 --sort          oriented \
 --outputModel   ID:ref_reordered \
END \
reindex \
 --inputModel    ID:deqDis \
 --sort          oriented \
 --outputModel   ID:dis_reordered \
END \
sample \
 --inputModel    ID:ref_reordered \
 --inputMap      /path/to/contents/basketball_player_fr%04d.png \
 --mode          grid \
 --useNormal \
 --useFixedPoint \
 --minPos        "-725.812988 -483.908997 -586.02002" \
 --maxPos        "1252.02002 1411.98999 1025.34998" \
 --bilinear \
 --gridSize      1024 \
 --hideProgress  1 \
 --outputModel   ID:ref_pc \
END \
sample \
 --inputModel    ID:dis_reordered \
 --inputMap      results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.png \
 --mode          grid \
 --useNormal \
 --useFixedPoint \
 --minPos        "-725.812988 -483.908997 -586.02002" \
 --maxPos        "1252.02002 1411.98999 1025.34998" \
```

```
    --bilinear \
    --gridSize      1024 \
    --hideProgress  1 \
    --outputModel   ID:dis_pc \
   END \
   compare \
    --mode          pcc \
    --inputModelA   ID:ref_pc \
    --inputModelB   ID:dis_pc \
    --resolution    1977.833008 \
    --outputCsv     results/F001/s3c1r2_bask/metric_pcc.csv \
   > results/F001/s3c1r2_bask/metric_pcc.log
NbOutputFaces      : 75648
TotalBitstreamBits : 150448
GridD1             : 73.833939
GridD2             : 75.434639
GridLuma           : 36.139542
GridChromaCb       : 43.351307
GridChromaCr       : 45.376358
IbsmGeom           : 46.775490
IbsmLuma           : 33.573721
EncTime            : 27.0134349
DecTime            : 0.34059222
```

The –tmmMetric paramerter executes the vmesh metric software to compute metrics rather than the mm software. In this case, the logs are as follows:

```
$ ./scripts/run.sh \
  --condId=1 \
  --seqId=3 \
  --rateId=2 \
  --cfgdir=generatedConfigFiles \
  --outdir=results \
  --tmmMetric
Encode: results/F001/s3c1r2_bask/s3c1r2_bask
./build/Release/bin/encode \
    --config=./generatedConfigFiles/s3c1r2_bask//encoder.cfg \
    --frameCount=1 \
    --compressed=results/F001/s3c1r2_bask/s3c1r2_bask.vmesh \
   > results/F001/s3c1r2_bask/encoder.log 2>&1
Decode: results/F001/s3c1r2_bask/s3c1r2_bask
./build/Release/bin/decode \
    --config=./generatedConfigFiles/s3c1r2_bask//decoder.cfg \
    --compressed=results/F001/s3c1r2_bask/s3c1r2_bask.vmesh \
    --decMesh=results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.obj \
    --decTex=results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.png \
    --decMat=results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.mtl \
   > results/F001/s3c1r2_bask/decoder.log 2>&1
Metrics: results/F001/s3c1r2_bask/s3c1r2_bask
./build/Release/bin/metrics \
    --config=./generatedConfigFiles/s3c1r2_bask//mmetric.cfg \
    --decMesh=results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.obj \
    --decTex=results/F001/s3c1r2_bask/s3c1r2_bask_%04d_dec.png \
    --frameCount=1 \
```

```
   > results/F001/s3c1r2_bask/metric_met.log
NbOutputFaces      : 75648
TotalBitstreamBits : 150448
GridD1             : 73.8339386
GridD2             : 75.434639
GridLuma           : 36.1395416
GridChromaCb       : 43.3513069
GridChromaCr       : 45.376358
IbsmGeom           : 46.7754899
IbsmLuma           : 33.5737214
EncTime            : 28.7190478
DecTime            : 0.355798779
```

5.5 Collect results

To collect the results from the log files (encoder, decoder and metric), the `./scripts/collect_results.sh` script can be uses:

```
$ ./scripts/collect_results.sh
./scripts/collect_results.sh Collect results from log files

  Usage:
    -h|--help   : print help
    -q|--quiet  : disable logs         (default: 1 )
    --condId=   : condition: 1, 2      (default: 1 )
    --seqId=    : seq: 1,2,3,4,5,6,7,8 (default: 1 )
    --rateId=   : Rate: 1,2,3,4,5      (default: 1 )
    --vdmc      : vdmc bitstream file  (default: "" )
    --logenc    : encoder log file     (default: "" )
    --logdec    : decoder log file     (default: "" )
    --logmet    : metrics log file     (default: "" )
    --csv       : generate .csv file   (default: "" )

  Examples:
    ./scripts/collect_results.sh -h
    ./scripts/collect_results.sh \
      --condId=1 \
      --seqId=3 \
      --rateId=3 \
      --vdmc=test.bin \
      --logenc=encoder.log \
      --logdec=decoder.log \
      --logmet=metric.log
```

This script can be used to parse the log files and display or get the bitrate/metric values:

```
$ ./scripts/collect_results.sh  \
  --condId  1 \
  --seqId   2 \
  --rateId  2 \
  --vdmc    s2c1r2_sold.vmesh \
  --logenc  encoder.log     \
  --logdec  decoder.log     \
  --logmet  metrics.log
2,1,2,152064,206976,72.1250986,74.0027083,29.7090586,43.1735896,43.2807054,\
```

```
48.3653428,29.1616112,24.8878219,0.257652824,484.457,178.211
$ RES=( $( ./scripts/collect_results.sh  \
  --condId  1 \
  --seqId   2 \
  --rateId  2 \
  --vdmc    s2c1r2_sold.vmesh \
  --logenc  encoder.log     \
  --logdec  decoder.log     \
  --logmet  metrics.log ) );
$ for((i=0;i<16;i++)); do printf "RES[%2d] = %s \n" $i ${RES[$i]}; done
RES[ 0] = 2
RES[ 1] = 1
RES[ 2] = 2
RES[ 3] = 152064
RES[ 4] = 206976
RES[ 5] = 72.1250986
RES[ 6] = 74.0027083
RES[ 7] = 29.7090586
RES[ 8] = 43.1735896
RES[ 9] = 43.2807054
RES[10] = 48.3653428
RES[11] = 29.1616112
RES[12] = 24.8878219
RES[13] = 0.257652824
RES[14] = 484.457
RES[15] = 178.211
```

5.6 Run all experiments and create render and graph pdf files

To run CTC experiments with all sequences, all conditions and all rates as definied in CTC conditions, the `./scripts/run_all.sh`script can be used :

```
$ ./scripts/run_all.sh --help
./scripts/run_all.sh execute all encoding/decoding/metrics
  Usage:
    -h|--help    : print help
    -q|--quiet   : disable logs                    (default: 1 )
    -f|--frames  : frame count                     (default: 1 )
    -c|--cfgdir  : configured directory            (default: "config" )
    -o|--outdir  : output directory                (default: tests )
    --experiments: csv configuration files         (default: test.csv )
    --tmmMetric  : Use TMM metric software         (default: 0 )
    -t|--threads : Number of parallel experiments (default: 1 )
    --render     : Create pdf with rendered images(default: 0 )
    --graph      : Create pdf with metric graphs  (default: 0 )
    --xlsm       : Create CTC xlsm files           (default: 0 )

  Examples:
    ./scripts/run_all.sh -h
    ./scripts/run_all.sh \
      --experiments ./scripts/test.csv \
      --outdir      experiments \
      --cfgdir      generatedConfigFilesHM \
      --frame       2 \
```

```
    --graph \
    --render \
    --xlsm \
    --quiet
```

This scripts executes severals experiments that must be defined in `./scripts/test.csv` files. This file defined the experiments that must be evaluated, one experiments by line. Each experiments must set:

- Name: the name of the experiment.
- EncParams: the encoder parameters used.
- DecParams: the decoder parameters used.

An example of this file is the following one:

```
$ cat ./scripts/test.csv
Name,EncParams,DecParams
anchor,,
texture1k,--textureVideoWidth=1024 --textureVideoHeight=1024,
texture2k,--textureVideoWidth=2048 --textureVideoHeight=2048,
```

The experiments can be executed with the following command line:

```
$ ./scripts/run_all.sh \
  --frame=4 \
  --threads 10 \
  --render \
  --graph \
  --xlsm \
  --quiet
```

The `--graph` and `--renderer` options create pdf files with the graph and the render images of all the experiments defined in `./scripts/test.csv`. Examples of the created pdf files can be seen in figures 2 and 3.

The `--xlsm` option fill the CTC XLSM spreadsheet with the results of the current experiences. The first line of the CSV file is set as anchor of the experiences and the other one are compared to the anchor and between them. With the previously presented CSV files teh following files are created:

- ./experiments/F004_anchor_vs_texture1k.xlsm
- ./experiments/F004_anchor_vs_texture2k.xlsm
- ./experiments/F004_texture1k_vs_texture2k.xlsm

Note: The `--xlsm` option uses `openpyxl` Python module to fill the XLSM files and requieres Python3 to work properly.

The `--threads N` option allows experiments to be run in parallel with N which defines the number of parallel tests.

Note: This option has been used on Linux and uses Linux commands to work. Please, use this script in a Linux terminal. On Window, please uses: msys, cygwin, mingw or Windows Subsystem for Linux (WSL).

5.7 View decoded sequences

The subjective quality of the decoded sequences can be evaluated by playing the decoded .ply/.png files with the mpeg-pcc-renderer (http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-renderer.git).

The following commands can be used to install and to execute this software:

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

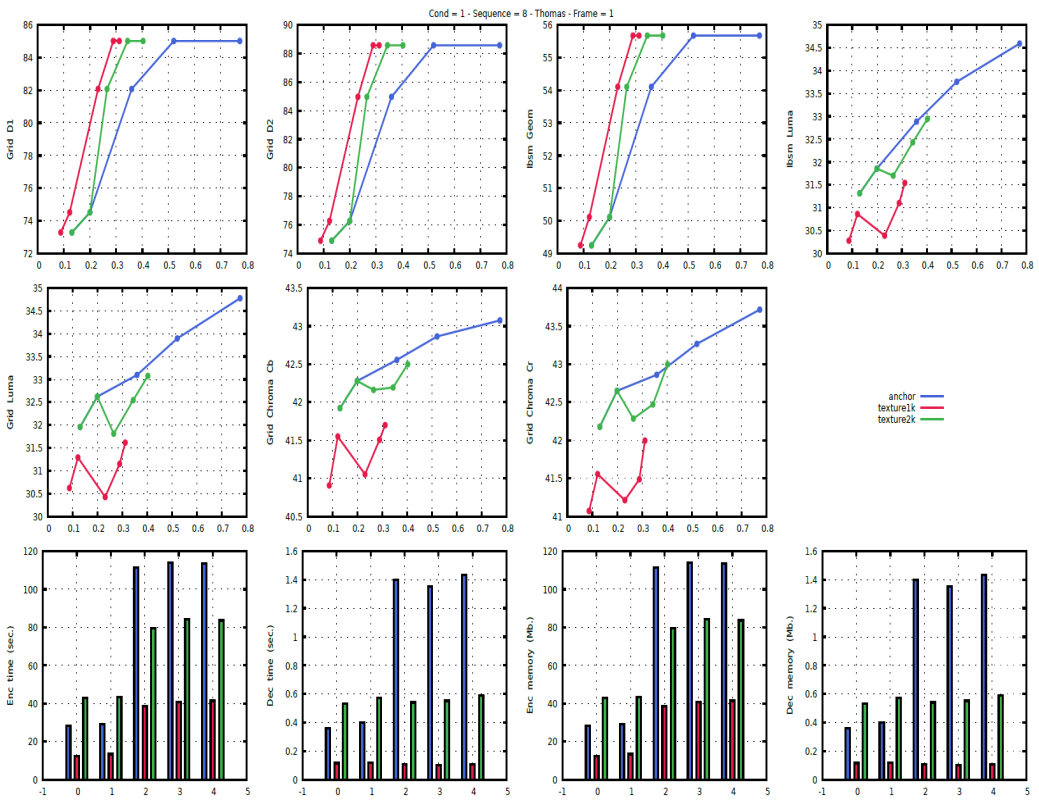

Figure 2 – Example of graphs.

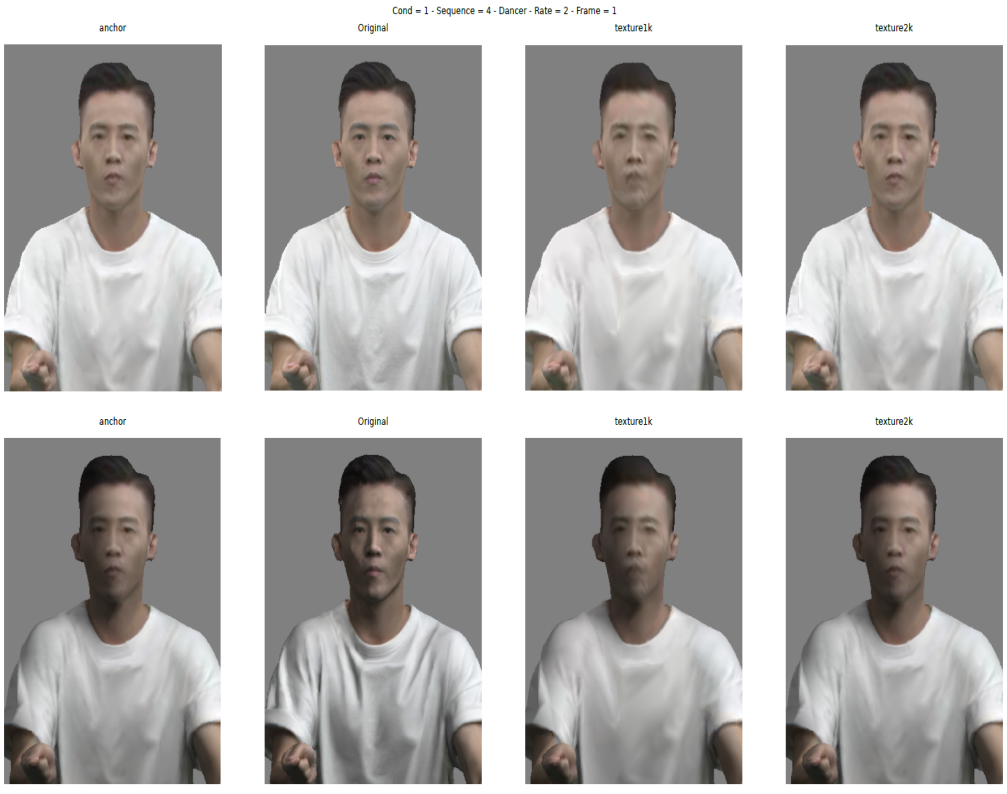

Figure 3 – Example of render images.

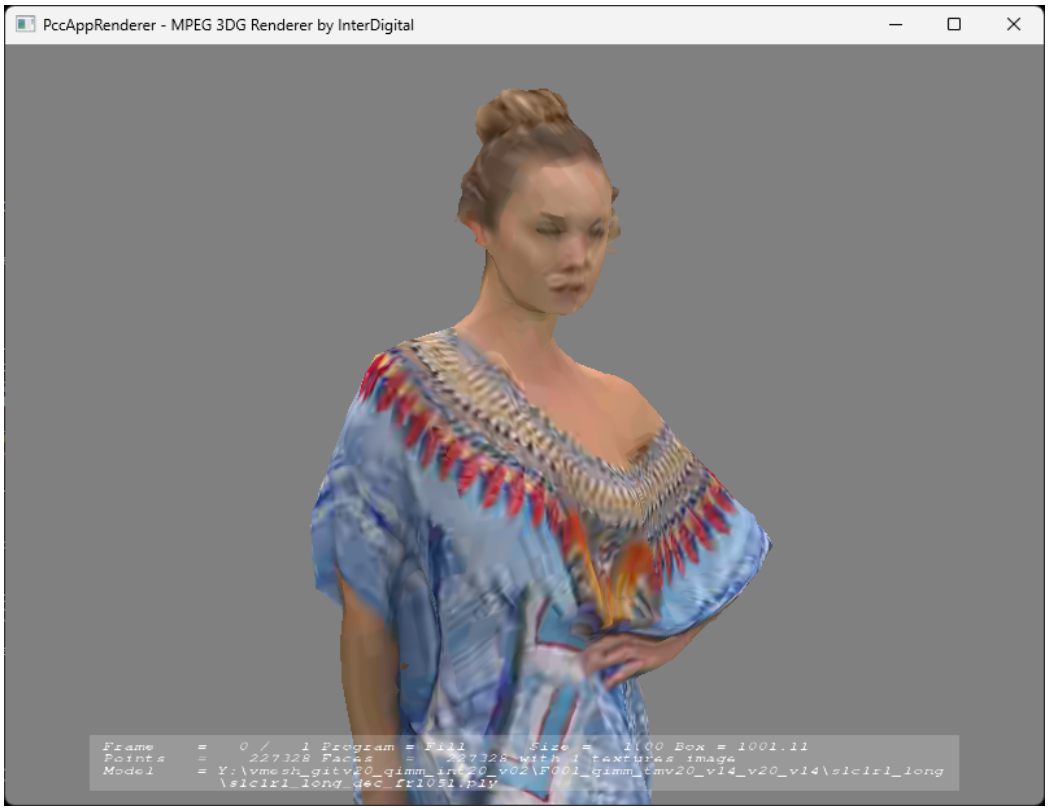

Figure 4 – Screenshoot of the mpeg-pcc-renderer software.

```
./build.sh
./bin/windows/Release/PccAppRenderer.exe \
  -f ./s1c1r1_long/s1c1r1_long_dec_fr1051.ply
```

A specific script can be used to create video of the decoded sequences like shown in figure 3:

```
./scripts/renderer.sh \
  -i ./s1c1r1_long/  \
  --videoType=4   \
  -w 600 \
  -h 800 \
  --cameraPathIndex=10
```

6 Main software input parameters

The following subsections contain input parameters for encoding, decoding, and metrics software.

6.1 Encode software input parameters

–key=value	**Usage**
Common	
–help=0	This help text
-c, –config=...	–config=... Configuration file name
-v, –verbose=0	–verbose=0 Verbose output
Input	
–srcMesh=""	Input mesh

–key=value	Usage
–srcTex=“”	Input texture (deprecated consider using srcAttrs instead)
–srcAttrs=“”	Input attributes path: a%d.png,b%d.png…
–videoAttributeTypes=“0”	Input video attribute type: 0(text_coord), 0(text_coord)…
–videoAttributeCount=0	number of attributes to be coded by video codec
–videoAttributeTypes=“0”	Input video attribute type: 0(text_coord), 0(text_coord)…
–textureMapCount=1	number of texture maps per mesh :default 1
–positionBitDepth=12	Input positions bit depth
–texCoordBitDepth=12	Input texture coordinates bit depth
–referenceAttributeIdx=0	Index of the reference attribute for consistent tiling inference
–submeshCount=1	number of submeshes :default 1
–tileCountGeometry=1	number of tiles with geometry information in the atlas data sub-bitstream
–submeshSegmentationType=0	Submesh Segmentation Method: default 0
–enableSignalledIds=0	enableSignalledIds
–segmentThreshold=100	Threshold used for triangle segmentation by area (when submeshCount:2)
–allowSubmeshOverlap=1	number of submeshes :default 1
–segmentByBaseMesh=0	segmentation by basemesh
–zipperingMethod=-1	zippering method: -1: deactivated, 0/1: per sequence (fixed or max value), 2: per frame (max value), 3: per sub-mesh (max value), 4: per border value, 5: using indexes 7:boundary connect
–zipperingThrehsold=5	zippering threhsold: default =0
–zipperingMethodForUnmatchedLoDs=0	zippering method for unmatched LoDs: 0: no correction, 1: remove vertices from unmatched LoDs, 2: add vertices in neighbouring submesh for unmatched LoDs
–increaseTopSubmeshSubdivisionCount=0	increase top submesh subdivision iteration count by 1 (default: off)
–zipperingMatchedBorderPointDeltaVarianceThreshold=0	Write deltas of zippering matched border points if their varience is under this value
–linearSegmentation=1	linear or mixed segmentation into submeshes : default 1 (linear)
–TileSubmeshMappingSEI=0 (default =0)	send the tile submesh mapping SEI = 0)
–startFrameIndex=0	First frame number
–frameCount=1	Number of frames
–framerate=30	Frame rate
Output	
–compressed=“”	Compressed bitstream
–recMesh=“”	Reconstructed mesh
–recAttrs=“”	Reconstructed attribute paths
–dequantizeUV=1	Dequantize texture coordinates of the reconstructed meshes
–recMat=“”	Reconstructed materials
–reconstructNormals=1	0:no Normals 1:local coordinate

–key=value	Usage
ProfileCodecGroupId	
–profileCodecGroupIdc=HEVC_Main10	Profile Codec Group Idc
General	
–keep=0	Keep intermediate files
–keepBaseMesh=0	Keep intermediate BaseMesh files
–keepVideo=0	Keep intermediate Video files
–keepPreMesh=0	Keep intermediate mesh files generated during the preprocessing
–maxLayersMinus1=-1	Maximum layers for spatial scalability (default: 2)
–deformForRecBase=1	deform for rec base
–useSubmeshFolder=1	use folders to store submeshes
–checksum=1	Compute checksum
–lossless=0	0:Lossy, 1:Lossless
Group of frames analysis	
–gofMaxSize=32	Maximum group of frames size
–analyzeGof=0	Analyze group of frames
Geometry decimate	
–target=0.125	Target triangle count ratio
–minCCTriangleCount=0	minimum triangle count per connected component
–minPosition="0,0,0"	Min position
–maxPosition="0,0,0"	Max position
Texture parametrization	
–textureParameterizationType=0	Texture Parameterization Type of UVATLAS (0) or ORTHO (1) or NONE (-1)
–orthoAtlasUseVertexCriteria=0	Use Vertex Criteria (DEFAULT:1)
–orthoAtlasUseSeedHistogram=1	Determine CC seed using histogram (DEFAULT:1)
–orthoAtlasStrongGradientThreshold=180	Strong Gradient Threshold (DEFAULT:180)
–orthoAtlasMaxCCAreaRatio=1	Minimum Connected Components Area Ratio (DEFAULT:1)
–orthoAtlasMaxNumFaces=2147483647	Max Num Faces (DEFAULT:inf)
–orthoAtlasEnableFaceClusterMerge=1	Face Cluster Merge (DEFAULT:0)
–orthoAtlasLambdaRDMerge=1	Lambda RD for merging (DEFAULT:1.0)
–orthoAtlasCheck2DConnectivity=1	Enforce 2D connectivity (DEFAULT:1)
–orthoAtlasAdjustNormalDirection=0	Adjust normal for projected patches (DEFAULT:0)
–orthoAtlasApplyReprojection=0	Reproject patches after checking overlapping (DEFAULT:0)
–orthoAtlasMergeSmallCC=0	Merge Small Connected Components (DEFAULT:0)
–orthoAtlasSmallCCFaceNum=1	Small CC face numbers (DEFAULT:1)
–orthoAtlasEnablePatchScaling=0	Enable patch scaling (DEFAULT:0)
–orthoAtlasEnablePatchTemporalStabilization=0	Enable patch temporal stabilization (DEFAULT:1)
–orthoAtlasDeriveTextCoordFromPos=2	Derivation of text. coords. from position: 0 (disabled), 1 (using UV coords), 2 (using FaceId, DEFAULT), 3 (using Connected Components)
–orthoAtlasUse45DegreeProjection=0	Use 45 degree projection (DEFAULT:1)
–orthoAtlasPackingScaling=0.95	Packing scaling adjustment (DEFAULT:0.9)
–orthoAtlasPackSmallPatchesOnTop=0	Pack small patches on top (DEFAULT:1)

–key=value	Usage
–orthoAtlasPackingType=0	Default Packing (0, default), Tetris Packing (1), Projection Packing (2), Projection Packing with Temporal Stabilization (3)
–orthoAtlasUpdateTextureSize=2	Update texture size from textureVideoWidth/textureVideoHeight for orthoAtlasPackingType:3
–orthoAtlasBiasEnable=0	Enable the transmission of sub-patch bias synatx elemnts DEFAULT:0
–orthoAtlasParameterAdjustment=1	Update subpatch position and size:1
–orthoAtlasLog2AnalysisBlockSize=0	Log 2 of the size of the block used for parameter adjustment:0
–orthoFaceIdPresentFlag=0	Use face ID (DEFAULT:false)
–useRawUV=0	Use RAW subpatch
–targetCountThreshold=5	targetCountThreshold for RAW subpatch
–rawTextcoordBitdepth=6	Bitdepth for texture coordinates in raw subpatch
–textureParametrizationQuality=DEFAULT	Quality level of DEFAULT, FAST or QUALITY
–textureParametrizationMaxCharts=0	Maximum number of charts to generate
–textureParametrizationMaxStretch=0.16667	Maximum amount of stretch 0 to 1
–textureParametrizationGutter=2	Gutter width betwen charts in texels
–textureParametrizationWidth=512	texture width
–textureParametrizationHeight=512	texture height
–textureBGR444=0	Texture video encoded in BGR444
–texturePlacementPerPatch=1	Texture of a submesh corresponding to a patch is placed within the area indicated by the patch
Geometry parametrization	
–baseIsSrc=0	Base models are src models
–subdivIsBase=0	Subdiv models are src models
–subdivInter=0	Subdiv inter
–subdivInterWithMapping=0	Subdiv inter with mapping
–encoderOptiFlag=0	0:Turn off the geometric optimization algorithm of the deformed mesh at the encoding end; 1: Turn on the geometric optimization algorithm of the deformed mesh at the encoding end
–maxAllowedD2PSNRLoss=1	Maximum allowed D2 PSNR Loss
–normalCalcModificationEnable=1	0: Calculate normal of cloudB from cloudA, 1: Use normal of cloudB(default)
–log10GeometryParametrizationQuantizationValue=5	Quantization value of the input of Geometry Parametrization expressed in log 10
–trackedMode=0	Tracked mode
Intra geometry parametrization	
–ai_sdeform=1	Apply deformation refinement stage
–ai_subdivIt=3	Subdivision iteration count
–ai_forceNormalDisp=0	Force displacements to aligned with the surface normals
–ai_unifyVertices=1	Unify duplicated vertices
–ai_deformNNCount=1	Number of nearest neighbours used during the initial deformation stage
–ai_deformNormalThres=0.1	Maximum allowed normal deviation during the initial deformation stage

–key=value	Usage
–ai_sampIt=3	Number of subdivision iterations used for geometry sampling
–ai_fitIt=16	Number of iterations used during the deformation refinement stage
–ai_smoothCoeff=0.25	Initial smoothing coefficient used to smooth the deformed mesh during deformation refinement
–ai_smoothDecay=0.75	Decay factor applied to intial smoothing coefficient after every iteration of deformation refinement
–ai_smoothMissedCoeff=0.1	Smoothing coefficient applied to the missed vertices
–ai_smoothMissedIt=10	Number of iterations when smoothing the positions of the missed vertices
–ai_smoothMethod=1	Smoothing method to be applied when smoothing the deformed mesh duringthe deformation refinement stage
–ai_deformUpdateNormals=1	Recompute normals after each iteration of deformation refinement
–ai_deformFlipThres=-0.5	Threshold to detect triangle normals flip
–ai_useInitialGeom=1	Use the initial geometry during the the deformation refinement stage
–ai_fitSubdiv=1	Update the positions of the decimated mesh to minimize displacements between the subdivided mesh and the deformed mesh
–ai_smoothMotion=1	Apply smoothing to motion instead of vertex positions
Inter geometry parametrization	
–ld_sdeform=1	Apply deformation refinement stage
–ld_subdivIt=3	Subdivision iteration count
–ld_forceNormalDisp=0	Force displacements to aligned with the surface normals
–ld_unifyVertices=1	Unify duplicated vertices
–ld_deformNNCount=1	Number of nearest neighbours used during the initial deformation stage
–ld_deformNormalThres=0.1	Maximum allowed normal deviation during the initial deformation stage
–ld_sampIt=3	Number of subdivision iterations used for geometry sampling
–ld_fitIt=16	Number of iterations used during the deformation refinement stage
–ld_smoothCoeff=0.25	Initial smoothing coefficient used to smooth the deformed mesh during deformation refinement
–ld_smoothDecay=0.75	Decay factor applied to intial smoothing coefficient after every iteration of deformation refinement
–ld_smoothMissedCoeff=0.1	Smoothing coefficient applied to the missed vertices
–ld_smoothMissedIt=10	Number of iterations when smoothing the positions of the missed vertices
–ld_smoothMethod=1	Smoothing method to be applied when smoothing the deformed mesh duringthe deformation refinement stage

–key=value	Usage
–ld_deformUpdateNormals=1	Recompute normals after each iteration of deformation refinement
–ld_deformFlipThres=-0.5	Threshold to detect triangle normals flip
–ld_useInitialGeom=1	Use the initial geometry during the the deformation refinement stage
–ld_fitSubdiv=1	Update the positions of the decimated mesh to minimize displacements between the subdivided mesh and the deformed mesh
–ld_smoothMotion=1	Apply smoothing to motion instead of vertex positions
Subdivision	
–subdivisionIterationCount=3	Subdivision iteration count
–interpolateSubdividedNormalsFlag=0	Interpolate subdivided normals on edges if true
–subdivisionEdgeLength=0	Subdivision edge length threshold
Quantization	
–log2LevelOfDetailInverseScale="1,1,1"	Quantization LoD inverse scale for displacements
–liftingQP="16,28,28"	Quantization parameter for displacements
–adaptiveScale="110,110,110"	Scale factor for displacements
–liftingBias="0.333333,0.333333,0.333333"	Quantization bias for displacements
–lodDisplacementQuantizationFlag=0	Use quantization parameter per LoD for displacements
–liftingQP2="-1143724448,32744,1232715136,21920,0,0,0,0,0"	Quantization parameter for displacements
-l, –lodAdaptiveSubdivisionFlag=0	–lodAdaptiveSubdivisionFlag=0 toggle lodAdaptiveSubdivision (Default: false)
–sm=0	Subdivision method for all iterations (0: MIDPOINT (default), 1: LOOP , 2: NORMAL, 3: PYTHAG)
–sml0=0	Subdivision method for the 1st iteration (0: MIDPOINT (default), 1: LOOP , 2: NORMAL, 3: PYTHAG)
–sml1=0	Subdivision method for the 2nd iteration (0: MIDPOINT (default), 1: LOOP , 2: NORMAL, 3: PYTHAG)
–sml2=0	Subdivision method for the 3rd iteration (0: MIDPOINT (default), 1: LOOP , 2: NORMAL, 3: PYTHAG)
–bitDepthOffset=0	Quantization parameter for displacements
Lifting	
–transformMethod=1	Selecting the transform method
–liftingAdaptivePredictionWeightFlag=0	Enable Adaptive Lifting Prediction Weight
–liftingAdaptiveUpdateWeightFlag=1	Enable signalling of lifting update parameters per LoD
–liftingValenceUpdateWeightFlag=1	Enable Valence Lifting Update Weight
–liftingAdaptiveUpdateWeightNumerator="63,21,7,1"	The adaptive lifting update weight numerators
–liftingAdaptiveUpdateWeightDenominatorMinus1="79,19,4,0"	The adaptive lifting update weight Denominators

–key=value	Usage
–liftingAdaptivePredictionWeightNumerator="1,1,1,1"	The adaptive lifting prediction weight numerators
–liftingAdaptivePredictionWeightDenominatorMinus1="2,1,0,1"	The adaptive lifting prediction weight Denominators
–InverseQuantizationOffsetFlag=1	Inverse quantization offset enabling flag
–dirliftScale1=950	Directional lifting scale 1
–dirliftDeltaScale2=25	Directional lifting scale2-scale1
–dirliftDeltaScale3=20	Directional lifting scale3-scale2
–dirliftScaleDeno=999	Directional lifting scale Deno
–dirlift=0	Enable directional lifting
Base mesh	
–baseMeshPositionBitDepth=10	Quantization bits for base mesh positions
–baseMeshTexCoordBitDepth=8	Quantization bits for base mesh texture coordinates
–invertOrientation=0	Invert triangles orientation
–unifyVertices=0	Unify duplicated vertices
–reverseUnification=0	Reverse the unified vertices
–meshCodecId=1	Mesh codec id
–baseMeshVertexTraversal="degree"	EB vertex traversal method: eb: use edge breaker traversal for vertex traversal during predictions degree: use vertex degree traversal during predictions
–baseMeshDeduplication=0	base mesh deduplicate positions and attributes
–entropyPacket=0	entropy packet
–dracoUsePosition=0	Draco use position
–dracoUseUV=0	Draco use UV
–dracoMeshLossless=0	draco mesh lossless
–profileGeometryCodec=0	activate base mesh codec profiling (DEFAULT:0), set to n $>$ 0 to activate and perform n runs of the codec for precise timings
Motion	
–motionGroupSize=16	Motion field coding vertices group size, 1
–motionWithoutDuplicatedVertices=0	Motion field coding by integrating duplicated vertices in reference frames
–motionVertexTraversal="degree"	Vertex traversal method for motion predictions: default: use base mesh default order for vertex traversal during predictions degree: use vertex degree traversal during predictions
Geometry video	
–geometryVideoCodecId=HM	Geometry video codec id
–geometryVideoEncoderConfig=""	Geometry video config file
–lodPatchesEnable=0	Displacements packing in video mode:
0: all LoD displacement in one patch	all LoD displacement in one patch
(asve_lod_patches_enable_flag =	(asve_lod_patches_enable_flag == 0) 1: LoD displacement per patch
(asve_lod_patches_enable_flag =	== 1)
–geometryVideoBitDepth=10	Geometry video bitdepth
–displacementVideoChromaFormat=1	Chroma Format of displacement : 0.400YUVp 1.420YUVp 2.422YUVp 3.444YUVp

–key=value	Usage
–log2GeometryVideoBlockSize=6	log2 of the blocksize(width=height) for the geometry video
Displacements	
–encodeDisplacements=2	Displacements coding mode: 0: no displacements coding 1: arithmetic coding 2: video coding
–applyOneDimensionalDisplacement=1	Apply one dimensional displacement
–applyLiftingOffset=1	Apply lifting offset
–displacementFillZerosThreshold=0.5	Displacement fill zeros threshold
–displacementFillZerosFlag=0	Displacement fill zeros flag
–displacementReversePacking=1	Displacement reverse packing
–displacementQuantizationType=0	Displacement packing type: 0-DEFAULT, 1-ADAPTIVE
–subBlocksPerLoD="1,1,1,1"	number of subblocks per level of detail for arithmetic coding
–LoDSubdiv="0,0,1,1"	Specifies LoD subdivision to blocks
Transfer texture	
–textureTransferEnable=1	Texture transfer enable
–textureTransferSamplingSubdivisionIterationCount=3	Texture transfer sampling subdivision iteration count
–textureTransferPaddingBoundaryIterationCount=2	Texture transfer padding boundary iteration count
–textureTransferPaddingDilateIterationCount=2	Texture transfer padding dilate iteration count
–textureTransferPaddingMethod=smoothed_push_-pull	Texture transfer padding method:
–textureTransferPaddingSparseLinearThreshold=0.5	Texture transfer padding sparse linear threshold
–textureTransferBasedPointcloud=1	Texture transfer padding sparse linear threshold
–textureTransferPreferUV=0	Texture transfer prefer UV
–textureTransferWithMap=0	Texture transfer with map for reconstructe sampling
–textureTransferWithMapSource=0	Texture transfer with map for source sampling
–textureTransferMapSamplingParam=1	Texture transfer map sampling param
–textureTransferMethod=0	Texture transfer method: 0: pcc 1: simple 2: simple new, 3: optimized
–textureTransferGridSize=0	textureTransferGridSize
–textureTransferMapProjDim=0	textureTransferMapProjDim
–textureTransferSigma=0.2	textureTransferSigma
–textureTransferMapNumPoints=8	textureTransferMapNumPoints
–textureTransferCopyBackground=1	textureTransferMapNumPoints
Motion coding	
–maxNumNeighborsMotion=3	Max number of vertex neighbors in motion coding
–maxNumMotionVectorPredictor=3	Max number of motion vector predictor in motion coding
Texture video	
–encodeTextureVideo=1	Encode texture video
–textureVideoCodecId=HM	Texture video codec id
–textureVideoEncoderConfig=""	Texture video encoder configuration file
–textureVideoEncoderConvertConfig=""	HDRTools encode configuration file
–textureVideoDecoderConvertConfig=""	HDRTools decode configuration file
–textureVideoDownsampleFilter=4	Chroma downsample filter in [0;22]

–key=value	Usage
–textureVideoUpsampleFilter=0	Chroma upsample filter in [0;7]
–textureVideoFullRange=0	Texture video range: 0: limited, 1: full
–textureVideoQP=8	Quantization parameter for texture video
–textureVideoWidth=2048	Output texture width
–textureVideoHeight=2048	Output texture height
–textureInputBitDepth=8	Input Texture map bitdepth
–textureVideoBitDepth=10	Texturemap Video bitdepth
–useOccMapRDO=1	Use Occupancy maps in the RDO process in HM
–occMapFilename="_occmap.yuv"	Occupancy maps Filename
–jointTextDisp=0	joint coding of texture and displacement
–scalableEnableFlag=0	Texture map scalable flag
–attributeExtractionEnable=0	enable attribute extraction, 0: disable (default), 1: enable
–attributeTransformParamsV3CSEIEnable=0	transformation params SEI for v3c
–attributeTransformParamsBaseMeshSEIEnable=0	transformation params SEI for basemesh
Bitstreams	
–forceSsvhUnitSizePrecision=0	force SampleStreamV3CUnit size precision bytes
–meshCodecSpecificParametersInBMSPS=0	enable storage of mesh codec specific parameters in BMSPS, 0: disable (default), 1: enable
–motionCodecSpecificParametersInBMSPS=0	enable storage of motion codec specific parameters in BMSPS, 0: disable (default), 1: enable
Metrics	
–pcc=0	Compute pcc metrics
–ibsm=0	Compute ibsm metrics
–pcqm=0	Compute pcqm metrics
–gridSize=1024	Grid size
–resolution=0	Resolution
–pcqmRadiusCurvature=0.001	PCQM radius curvature
–pcqmThresholdKnnSearch=20	PCQM threshold Knn search
–pcqmRadiusFactor=2	PCQM radius factor
Caching	
–cachingDirectory=""	Caching directory
–cachingPoint=none	Caching points: - 0/none : off - 1/simplify: symplify - 2/textgen : textgen - 3/subdiv : subdiv - 255/create: create caching files
Coding Structure	
–GOPSize=0	GOP Size
Normals	
–encodeNormals=0	Whether to encode the normals: Default: 0
–normalInt=1	Are the normals integer (1) or float (0): Default: 1
–normalBitDepth=16	Input Normals bit depth, Default: 16. If the input Normals bit depth is larger than normalBitDepth value, then input Normals would be clamp to normalBitDepth
–baseMeshNormalBitDepth=16	Quantization bits for base mesh Normals, Default: 16
–minNormals="-1,-1,-1"	Min Normals, Default: { -1.0, -1.0, -1.0 }

–key=value	Usage
–maxNormals="1,1,1"	Max Normals, Default: { 1.0, 1.0, 1.0 }
–predNormal=3	Prediction scheme for Normals in EB. 1: Delta, 2: MPARA, 3: Cross, Default: 3
–normalsOctahedral=1	Whether to encode normals using 2D Octahedral representation: Default: 1
–qpOctahedral=16	Quantization bits for 2D Normals representation, Default: 16
–Frame1=P	1 0 0 0
–Frame2=P	1 0 0 0
–Frame3=P	1 0 0 0
–Frame4=P	1 0 0 0
–Frame5=P	1 0 0 0
–Frame6=P	1 0 0 0
–Frame7=P	1 0 0 0
–Frame8=P	1 0 0 0
–Frame9=P	1 0 0 0
–Frame10=P	1 0 0 0
–Frame11=P	1 0 0 0
–Frame12=P	1 0 0 0
–Frame13=P	1 0 0 0
–Frame14=P	1 0 0 0
–Frame15=P	1 0 0 0
–Frame16=P	1 0 0 0
–Frame17=P	1 0 0 0
–Frame18=P	1 0 0 0
–Frame19=P	1 0 0 0
–Frame20=P	1 0 0 0
–Frame21=P	1 0 0 0
–Frame22=P	1 0 0 0
–Frame23=P	1 0 0 0
–Frame24=P	1 0 0 0
–Frame25=P	1 0 0 0
–Frame26=P	1 0 0 0
–Frame27=P	1 0 0 0
–Frame28=P	1 0 0 0
–Frame29=P	1 0 0 0
–Frame30=P	1 0 0 0
–Frame31=P	1 0 0 0
–Frame32=P	1 0 0 0
–Frame33=P	1 0 0 0
–Frame34=P	1 0 0 0
–Frame35=P	1 0 0 0
–Frame36=P	1 0 0 0
–Frame37=P	1 0 0 0
–Frame38=P	1 0 0 0
–Frame39=P	1 0 0 0
–Frame40=P	1 0 0 0
–Frame41=P	1 0 0 0
–Frame42=P	1 0 0 0
–Frame43=P	1 0 0 0
–Frame44=P	1 0 0 0
–Frame45=P	1 0 0 0

–key=value	Usage
–Frame46=P	1 0 0 0
–Frame47=P	1 0 0 0
–Frame48=P	1 0 0 0
–Frame49=P	1 0 0 0
–Frame50=P	1 0 0 0
–Frame51=P	1 0 0 0
–Frame52=P	1 0 0 0
–Frame53=P	1 0 0 0
–Frame54=P	1 0 0 0
–Frame55=P	1 0 0 0
–Frame56=P	1 0 0 0
–Frame57=P	1 0 0 0
–Frame58=P	1 0 0 0
–Frame59=P	1 0 0 0
–Frame60=P	1 0 0 0
–Frame61=P	1 0 0 0
–Frame62=P	1 0 0 0
–Frame63=P	1 0 0 0
–Frame64=P	1 0 0 0

6.2 Decode software input parameters

–key=value	Usage
Common	
–help=0	This help text
-c, –config=...	–config=... Configuration file name
-v, –verbose=0	–verbose=0 Verbose output
Input	
–compressed=""	Compressed bitstream
–decTargetLoD=-1	Target LoD
–decTargetLayer=2	Target Layer (default: 2)
–HMMCTSExtractorPath=""	HM MCTS Extractor Path: {MAINDIR}/externaltools/hm-16.21+scm-8 .8/bin/MCTSExtractorStatic
Output	
–decMesh=""	Decoded mesh
–decTex=""	Decoded texture
–decMat=""	Decoded materials
–dequantizeUV=1	Dequantize texture coordinates of the decoded meshes
–startFrameIndex=0	First frame number
–framerate=30	Frame rate
–reconstructNormals=1	0:No normals 1:local coordinate
General	
–keep=0	Keep intermediate files
–keepBaseMesh=0	Keep intermediate BaseMesh files

–key=value	Usage
–keepVideoFiles=0	Keep intermediate Video files
–checksum=1	Compute checksum
Decoder	
–textureVideoDecoderConvertConfig=“”	HDRTools decode cfg
–textureVideoUpsampleFilter=0	Chroma upsample filter in [0;7]
–textureVideoFullRange=0	Texture video range
Metrics	
–pcc=0	Compute pcc metrics
–ibsm=0	Compute ibsm metrics
–pcqm=0	Compute pcqm metrics
–gridSize=1024	Grid size
–resolution=0	Resolution
–minPosition=“0,0,0”	Min position
–maxPosition=“0,0,0”	Max position
–positionBitDepth=12	Position bit depth
–texCoordBitDepth=13	Texture coordinate bit depth
–pcqmRadiusCurvature=0.001	PCQM radius curvature
–pcqmThresholdKnnSearch=20	PCQM threshold Knn search
–pcqmRadiusFactor=2	PCQM radius factor
–srcMesh=“”	Metric Source mesh path
–srcTex=“”	Source texture path
–frameCount=0	Frame count

6.3 Metrics software input parameters

–key=value	Usage
Common	
–help=0	This help text
-c, –config=...	–config=... Configuration file name
-v, –verbose=0	–verbose=0 Verbose output
Source	
–srcMesh=“”	Source mesh
–srcTex=“”	Source texture
Decoded	
–decMesh=“”	Reconstructed/decoded mesh
–decTex=“”	Reconstructed/decoded texture
Sequence	
–startFrameIndex=1	First frame number
–frameCount=1	Number of frames
–minPosition=“0,0,0”	Min position
–maxPosition=“0,0,0”	Max position
–positionBitDepth=12	Position bit depth
–texCoordBitDepth=13	Texture coordinate bit depth
–dequantizeUV=1	Texture coordinates of the decoded meshes are quantized

–key=value	Usage
PCC metric	
–pcc=0	Compute pcc metrics
–gridSize=1024	Grid size
–resolution=0	Resolution
IBSM metric	
–ibsm=0	Compute ibsm metrics
PCQM metric	
–pcqm=0	Compute PCQM metrics
–pcqmRadiusCurvature=0.001	PCQM radius curvature
–pcqmThresholdKnnSearch=20	PCQM threshold Knn search
–pcqmRadiusFactor=2	PCQM radius factor

7 Other informations

7.1 Licence

POSSIBILITY OF SUCH DAMAGE.

7.2 Documentation

A pdf version of the user manual of the TMM can be found in the mpeg-vmesh-tm repository: (https://git.mpeg.expert/MP
dmc/software/mpeg-vmesh-tm/-/tree/main/doc/mpeg-vmesh-sw-manual.pdf).

7.3 Issue reporting

For any issues or questions don't hesitate to open issues in V-Mesh git repository or to contact us:

- Wenjie Zou (wjzou@xidian.edu.cn)
- Jungsun Kim (jungsun_kim@apple.com)

Bugs should be reported on the issue tracker at: (https://git.mpeg.expert/MPEG/3dgh/v-dmc/software/mpeg-vmesh-tm/-/issues).