Instructions

Building

OSX

- mkdir build
- cd build
- cmake .. -G Xcode
- open the generated xcode project and build it

Linux

- mkdir build
- cd build
- cmake ..
- make

Windows

- md build
- cd build
- cmake .. -G "Visual Studio 15 2017 Win64"
- open the generated visual studio solution and build it

Running

This TMC13 codec implementation encodes frame sequences. A single binary contains the encoder and decoder implementation, with selection using the `--mode` option. Documentation of options is provided via the `--help` command line option.

Runtime configuration and configuration files

All command line parameters may be specified in a configuration file.
A set of configuration file templates compliant with the current Common Test Conditions is provided in the cfg/ directory.

Example

To generate the configuration files, run the gen-cfg.sh script:

```
mpeg-pcc-tmc13-PDE/cfg$ ../scripts/gen-cfg.sh --all
```

An example script (`scripts/Makefile.tmc13-step`) demonstrates how to launch the encoder, decoder and metric software for a single input frame. The VERBOSE=1 make variable shows the detailed command execution sequence. Further documentation of the parameters are contained within the script.

The following example encodes and decodes frame 0100 of the sequence `Ford_01_q_1mm`, making use of the configuration file `cfg/lossy-geom-no-attrs/ford_01_q1mm/r01/encoder.cfg` and storing the intermediate results in the output directory `experiment/lossy-geom-no-attrs/ford_01_q1mm/r01/`.

```
mpeg-pcc-tmc13$ make -f $PWD/scripts/Makefile.tmc13-step \
    -C experiment/lossy-geom-no-attrs/ford_01_q1mm/r01/ \
    VPATH=$PWD/cfg/octree-predlift/lossy-geom-no-attrs/ford_01_q1mm/r01/ \
    ENCODER=$PWD/build/tmc3/tmc3 \
    DECODER=$PWD/build/tmc3/tmc3 \
    PCERROR=/path/to/pc_error \
    SRCSEQ=/path/to/Ford_01_q_1mm/Ford_01_vox1mm-0100.ply \
    NORMSEQ=/path/to/Ford_01_q_1mm/Ford_01_vox1mm-0100.ply

  [encode]  Ford_01_vox1mm-0100.ply.bin <- /path/to/Ford_01_q_1mm/Ford_01_vox1mm-
0100.ply
  [md5sum]  Ford_01_vox1mm-0100.ply.bin.md5
  [md5sum]  Ford_01_vox1mm-0100.ply.bin.ply.md5
  [decode]  Ford_01_vox1mm-0100.ply.bin.decoded.ply <- Ford_01_vox1mm-
0100.ply.bin
  [md5sum]  Ford_01_vox1mm-0100.ply.bin.decoded.ply.md5
  [metric]  Ford_01_vox1mm-0100.ply.bin.decoded.ply.pc_error <- Ford_01_vox1mm-
0100.ply.bin.decoded.ply
```

Another example

The example above is given by MPEG, which is a little complex. You can directly run my script `Encoder_Decoder_test/test.py` to test my code. Under near-lossless condition, `longdress_vox10_1300.bin` is 1010KB, while the original point cloud `longdress_vox10_1300.ply` is 23461KB.

longdress_vox10_1300.ply:

longdress_vox10_1300_dec.ply: