# Introduction

This document goes through all steps to generate point cloud data from input sources for 3GPP Beyond\_2D studies.

# Sources and prerequisites

## Content sources

Here is the list of all content sources with the provider and the content type provided with the corresponding links to download the content.

On Aspera website, 3GPP members can request the password by contacting 3gpp\_b2d\_datasets@interdigital.com.

|  |  |  |  |
| --- | --- | --- | --- |
| Content  Name | Content  Provider | Content type provided | Links to  source content |
| Mitch | Volucap | Mesh | <https://aspera.pub/I4tSQ8k> |
| JuggleSoccer | XD Productions | Mesh | <https://aspera.pub/I4tSQ8k> |
| Henry | RenderPeople | Blender project | [Henry 4D 007 Stretching | RENDERPEOPLE](https://renderpeople.com/3d-people/henry-4d-007-stretching/)  (Please purchase the blender project) |
| Nathalie | Volucap | Mesh | <https://aspera.pub/I4tSQ8k> |
| Aliyah | RenderPeople | Blender project | <https://download.renderpeople.com/free/rp_aliyah_4d_004_dancing_BLD.zip> |

## Prerequisites

If the Blender project is provided (Henry and Aliyah), the initial meshes will be generated using Blender software [1]. For detailed information, refer to Section III: Generation.

To use this framework, the user will need to install the following softwares:

* Python 3.11
* CMAKE 3.26 or above (but under 4.0.)
* A bash command line tool to run the scripts (on Windows, GitBash or MobaXTerm for example)

Please use a [python virtual environment](https://docs.python.org/3/library/venv.html#creating-virtual-environments) to install dependencies and run the scripts. A requirements.txt file is provided such that a suitable virtual environment can be set-up as follows:

python3 -m venv venv

venv\Scripts\activate # on Windows

. venv/bin/activate # on Linux

python -m pip install –upgrade pip

pip install -r requirements.txt

# Meshes generation from Blender

Open the Blender project for each corresponding sequence (Henry and Aliyah) and export the sequence to meshes following these steps:

1. Click on File menu, then “Export”, and “Wavefront (.obj)”
2. On the preset menu:

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated | In the Geometry section, untick “Normals”  In the Materials section, tick “Export” with Path Mode set to “Copy”  In the Animation section, tick “Export” to do the whole sequence |

1. Choose a specific output folder, for instance “mesh\_obj”

All md5 on meshes are provided in the ply\_generation/output\_info directory to ensure the right generation of meshes.

Once meshes data are available, pointcloud files generation is ready to be done, see details in section IV.

# Point Clouds generation from meshes

To proceed with the generation, the user needs to navigate to the ply\_generation/ directory, which contains:

* \*.py : Python scripts for generating PLY (pointcloud data) files.
* output\_info/ : Directory containing all expected md5sum result files for meshes (\*\_mesh\_md5.txt) and PLY files (\*\_output.log) for each content.
* jsons/ : Directory with an example of input configuration files.

## Test preparation

A JSON file named 3gpp\_selection.json is provided as input and is located in the jsons/ directory. It contains all information explained below in Table 1. This file should be updated for each sequence with the correct paths to the meshes for your environment (MeshObjPath and MeshTxtPath).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sequence | QP | Ratio | 1st Frame Index | Frame Number |
| Mitch | 11 | 0.70 | 1 | 475 |
| JuggleSoccer | 11 | 1 | 0 | 125 |
| Henry | 11 | 0.75 | 1 | 733 |
| Nathalie | 11 | 1 | 1 | 925 |
| Aliyah | 11 | 0.88 | 1 | 1112 |

Table 1 Input Configuration to generate point clouds with around 2 million points

## PLY generation

Once the Json file is updated with the correct meshes path, the PLY generation can be launched using the script exec\_ply\_generation.py which goes through the following steps:

1. **Dependencies installation**: It automatically downloads the mpeg-pcc-mmetric [2] in the output directory in the dependencies directory.
2. **Sampling pass**: This step gathers information on the sequence for quantifying the number of expected points. A ratio is provided via a Json file to ensure each sequence generates point clouds with around 2 million points per frame.
3. **Quantization pass**
4. **Cleaning pass**: This step removes all duplicate points using PyntCloud in Python.

It can be launched from your python environment with the following command:

python3 ply\_generation/exec\_ply\_generation.py -i ply\_generation/jsons/3gpp\_selection.json -o $YOUR\_OUTPUT\_PATH

for help on the script: python3 ply\_generation/exec\_ply\_generation.py -–help

usage: exec\_ply\_generation.py [-h] -i INPUTJSON -o OUTPUTDIR

This script generates point cloud frames (PLY) from meshes (OBJ+TXT)

options:

-h, --help show this help message and exit

-i INPUTJSON, --inputJson INPUTJSON

Json that contains the input meshes to process

-o OUTPUTDIR, --outputDir OUTPUTDIR

Output PLY directory

## Results’ check

In the output directory, you will find the generated PLY files and corresponding log files for each sequence.

To ensure the PLY generation proceeded as expected, md5 checksums (for meshes) and the number of points along with md5 checksums (for point clouds) are provided for each frame of each sequence. These details are compiled into a single file per sequence and stored in ply\_generation/output\_info.

# References

[1] Blender Software version 4.1.1 [blender.org - Home of the Blender project - Free and Open 3D Creation Software](https://www.blender.org/)

[2] mpeg-pcc-mmetric tag “1.1.7”

[MPEGGroup/mpeg-pcc-mmetric: MPEG PCC Mesh metric](https://github.com/MPEGGroup/mpeg-pcc-mmetric)