

📖 README.md

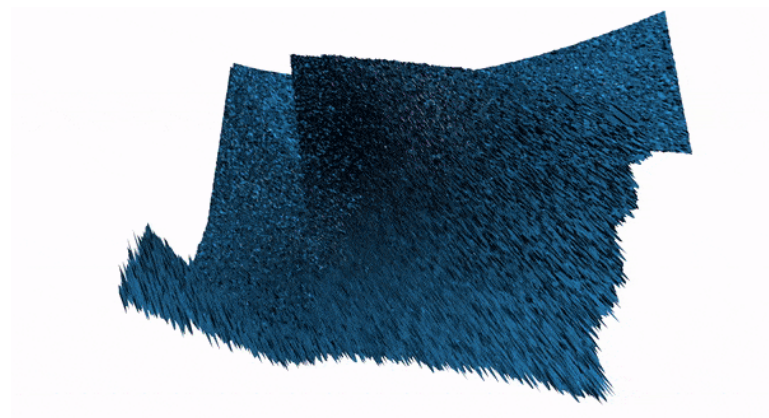
Organized Point Filters

API docs

This module is a collection of filters for use on **organized** point clouds (OPC). Note that this software has not been as extensively tested as my other work. The filters:

- Laplacian Mesh Smoothing applied to an implicit fully connected right cut triangular mesh of an OPC.
 - Single threaded, CPU Multi-threaded, and GPU accelerated.
- Bilateral Mesh Normal Smoothing applied to an implicit fully connected right cut triangular mesh of an OPC.
 - Single threaded, CPU Multi-threaded, and GPU accelerated.
- Intel RealSense Bilateral Spatial and Disparity Transform filters used on depth images.
 - I thought it would be useful to pull this code out of the Intel SDK such that it can be used by others who are not using realsense cameras. Apache 2.0 License.

Here is an example GIF of Laplacian and Bilateral Filtering of a noisy organized point cloud of stairs. The colors indicate the triangle normals of the mesh. The more uniform the colors, the smoother the surface



Installation

Installation is entirely through CMake now. You must have CMake 3.14 or higher installed and a C++ compiler with C++ 14 or higher. No built binaries are included currently.

Build Project Library

1. `mkdir cmake-build && cd cmake-build .` - create build folder directory
2. `cmake ../ -DCMAKE_BUILD_TYPE=Release .` For windows also add `-DCMAKE_GENERATOR_PLATFORM=x64`
3. `cmake --build . -j$(nproc)` - Build OPF

Build and Install Python Extension

1. Install [conda](#) or create a python virtual environment ([Why?](#)). I recommend conda for Windows users.
2. `cd cmake-build && cmake --build . --target python-package --config Release -j$(nproc)`
3. `cd lib/python_package && pip install -e .`

If you want to run the examples then you need to install the following (from main directory):

1. `pip install -r dev-requirements.txt`

You also need `cupy` to be installed with cuda device drivers if you want GPU acceleration. I cant vouch that this will always work:

1. `conda install cudatoolkit=10.1`
2. `pip install cupy-cuda101`

Documentation

Please see [documentation website](#) for more details.

Citation

To support our work please cite:

```
@article{s20174819,  
  author = {Castagno, Jeremy and Atkins, Ella},  
  title = {Polylidar3D - Fast Polygon Extraction from 3D Data},  
  journal = {Sensors},  
  volume = {20},  
  year = {2020},  
  number = {17},  
  article-number = {4819},  
  url = {https://www.mdpi.com/1424-8220/20/17/4819},  
  issn = {1424-8220}  
}
```