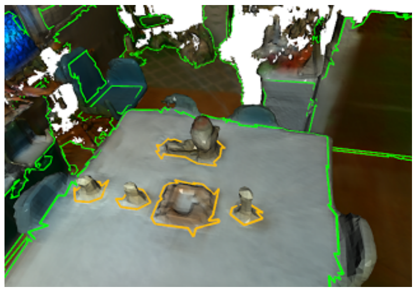
README.md

Polylidar3D

Polygon Extraction from 2D Point Sets, Unorganized/Organized 3D Point Clouds, and Triangular Meshes

Key Features • Documentation • Use Cases • Credits • Related • Citations • License



API docs | Cite 2D 10.1109-LRA.2020.3002212 | Cite 3D 10.3390/s20174819

Key Features

- Fast (Multi)Polygon Extraction from multiple sources of 2D and 3D Data
    - Written in C++ for portability
    - Extremely fast single-threaded but includes CPU multi-threading using data and task-based parallelism
    - Polygons with holes may be returned
- Python3 bindings using PyBind11
    - Low overhead for calling python/cpp interface (no copying of point cloud data)
- Python and C++ Examples
    - Examples from 2D point sets, unorganized 3D point clouds, organized 3D point clouds (i.e., range images), and user provided meshes
- Cross platform
    - Windows and Linux ready

Polylidar3D is a non-convex polygon extraction algorithm which takes as input either unorganized 2D point sets, unorganized 3D point clouds (e.g., airborne LiDAR point clouds), organized 3D point clouds (e.g., range images), or user provided meshes. In 3D, the non-convex polygons extracted represent flat surfaces in an environment, while interior holes represent obstacles on said surfaces. The picture above provides an examples of Polylidar3D extracting polygons from a 2D point set and a 3D triangular mesh; green is the concave hull and orange are interior holes. Polylidar3D outputs *planar* triangular segments and their polygonal representations. Polylidar3D is extremely fast, taking as little as a few milliseconds and makes use of CPU multi-threading and GPU acceleration when available.

Here is a small introductory blog-post about Polylidar3D.

Documentation and Branches

Please see documentation for installation, api, and examples. Note that Polylidar went though major changes in July 2020 for 3D work, now called `Polylidar3D`. The old repository for 2D work (and some *basic* 3D) is found in the branch polylidar2D and is connected to this paper. `Polylidar3D` can still handle 2D point sets but the API is different and not the focus of this repo. For papers referencing Polylidar2D and Polylidar3D please see Citations.

Eventually I am going to make a standalone cpp/header file for 2D point set -> polygon extraction for those that don't need any of the features of `Polylidar3D`.

Polylidar Use Cases

- Polylidar-RealSense - Live ground floor detection with Intel RealSense camera using Polylidar
- Polylidar-KITTI - Street surface and obstacle detection from autonomous driving platform
- PolylidarWeb. An very old Typescript (javascript) version with live demos of Polylidar2D
- Concave-Evaluation - Evaluates and benchmarks several competing concavehull algorithms

Credits

This software is only possible because of the great work from the following open source packages:

- Delaunator - Original triangulation library
- DelaunatorCPP - Delaunator ported to C++ (used)
- parallel-hashmap - Fast hashmap library (used)
- marl - A parallel thread/fiber task scheduler (used)
- PyBind11 - Python C++ Binding (used)
- Robust Geometric Predicates - Original Robust Geometric predicates
- Updated Predicates -Updated geometric predicate library (used)

Related Methods

2D ConcaveHull Extraction

- CGAL Alpha Shapes - MultiPolygon with holes
- PostGIS ConcaveHull - Single Polygon with holes
- Spatialite ConcaveHull - MultiPolygon with holes
- Concaveman - A 2D concave hull extraction algorithm for 2D point sets

Contributing

Any help or suggestions would be appreciated!

Citation

2D

If are using Polylidar for 2D work please cite:

J. Castagno and E. Atkins, "Polylidar - Polygons From Triangular Meshes," in IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 4634-4641, July 2020, doi: 10.1109/LRA.2020.3002212. Link to Paper

```
@ARTICLE{9117017,
  author={J. {Castagno} and E. {Atkins}},
  journal={IEEE Robotics and Automation Letters},
  title={Polylidar - Polygons From Triangular Meshes},
  year={2020},
  volume={5},
  number={3},
  pages={4634-4641}
}
```

3D

If you are using Polylidar3D for 3D work please cite:

J. Castagno and E. Atkins, "Polylidar3D - Fast Polygon Extraction from 3D Data," in MDPI Sensors, vol. 20, no.17, 4819, September 2020, doi: 10.3390/s20174819 Link to Paper

```
@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}
}
```

License

MIT

> GitHub @jeremybyu