

---

# **cilantro Documentation**

**Konstantinos Zampogiannis**

**Aug 15, 2023**



---

## Contents:

---

<b>1</b>	<b>About</b>	<b>3</b>
1.1	What is cilantro? . . . . .	3
1.2	Features . . . . .	3
<b>2</b>	<b>Getting started</b>	<b>7</b>
2.1	Dependencies . . . . .	7
2.2	How to build . . . . .	7







## 1.1 What is cilantro?

**cilantro** is a lean and fast C++ library for working with point cloud data, with emphasis given to the 3D case. It includes efficient implementations for a variety of common operations, providing a clean API and attempting to minimize the amount of boilerplate code. The library is extensively templated, enabling operations on point data of arbitrary numerical type and dimensionality (where applicable) and featuring a modular/extensible design of the more complex procedures, while, at the same time, providing convenience aliases/wrappers for the most common cases. A high-level description of **cilantro** can be found in our [technical report](#).

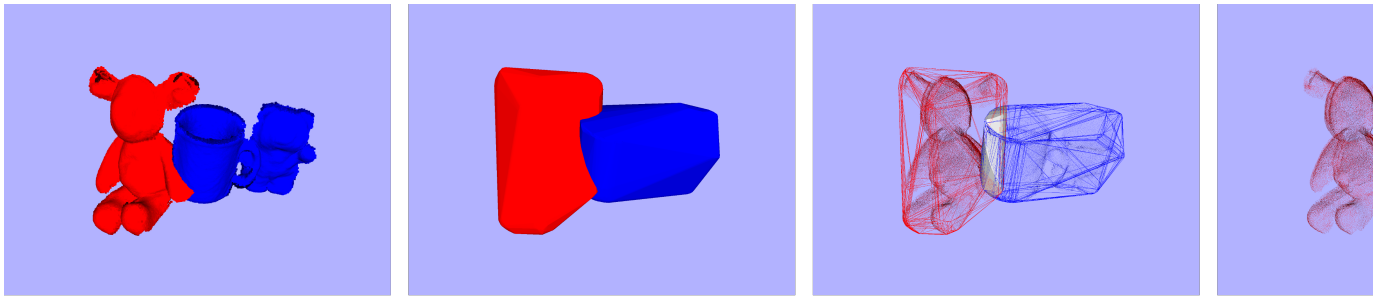
## 1.2 Features

### Basic operations:

- General dimension kd-trees (using bundled [nanoflann](#))
- Surface normal and curvature estimation from raw point clouds
- General dimension grid-based point cloud resampling
- Principal Component Analysis
- Basic I/O utilities for 3D point clouds (in PLY format, using bundled [tinyply](#)) and Eigen matrices
- RGBD images to/from point cloud utility functions

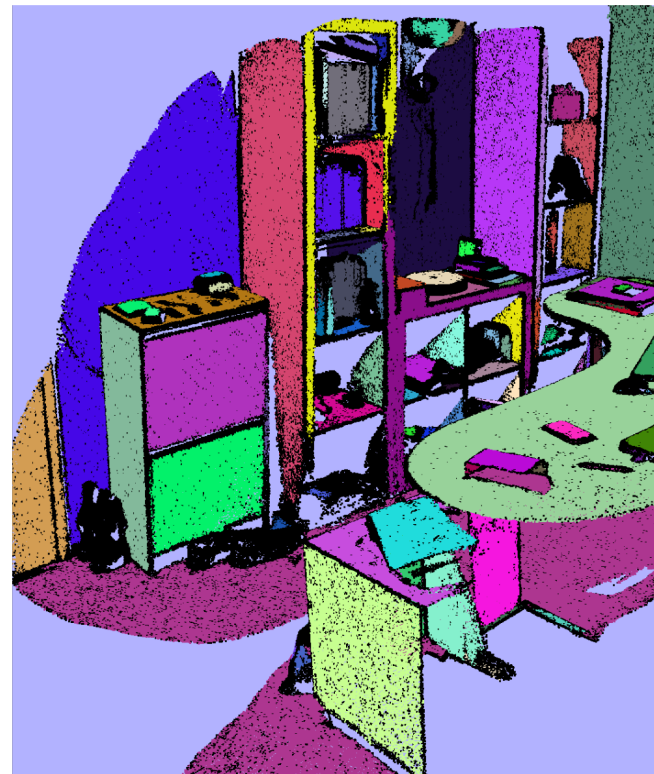
### Convex hulls and spatial reasoning tools:

- A general dimension convex polytope representation that is computed (using bundled [Qhull](#)) from either vertex or half-space intersection input and allows for easy switching between the respective representations
- A representation of generic (general dimension) space regions as unions of convex polytopes that implements set operations



### Clustering:

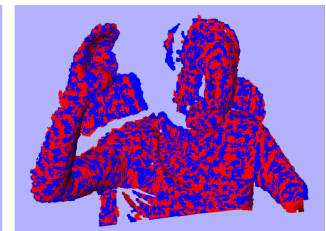
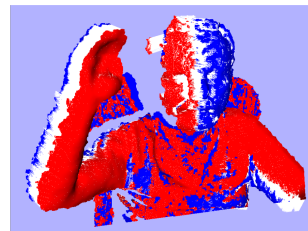
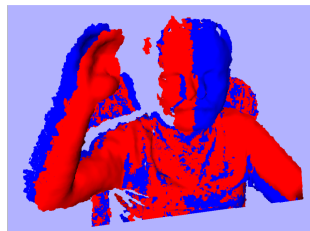
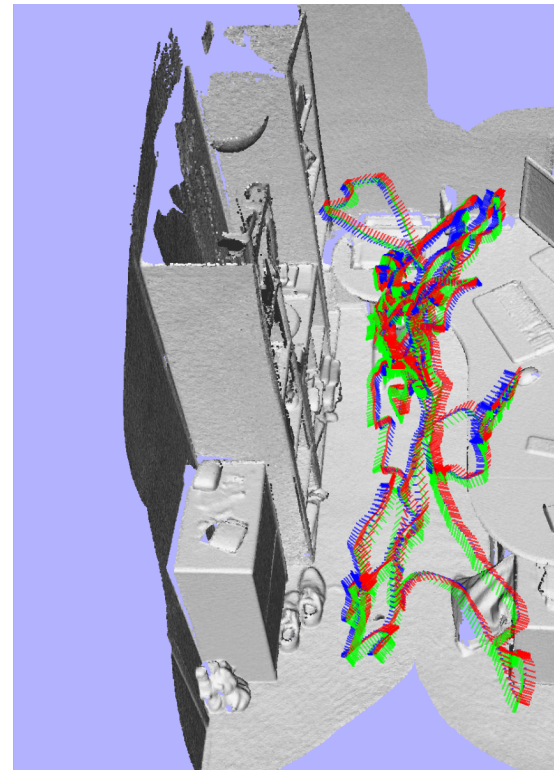
- General dimension k-means clustering that supports all distance metrics supported by nanoflann
- Spectral clustering based on various graph Laplacian types (using bundled Spectra)
- Mean-shift clustering with custom kernel support
- Connected component based point cloud segmentation that supports arbitrary point-wise similarity functions



### Geometric registration:

- Multiple generic Iterative Closest Point implementations that support arbitrary correspondence search methods in arbitrary point feature spaces for:
  - **Rigid** or **affine** alignment under the point-to-point metric (general dimension), point-to-plane metric (2D or 3D), or any combination thereof
  - **Non-rigid** alignment of 2D or 3D point sets, by means of a robustly regularized, **locally-rigid** or **locally-affine** deformation field, under any combination of the point-to-point and point-to-plane metrics; implementations for both *densely* and *sparsely* (by means of an Embedded Deformation Graph) supported warp fields are provided





#### Robust model estimation:

- A RANSAC estimator template and instantiations thereof for general dimension:
  - Robust hyperplane estimation
  - Rigid point cloud registration given noisy correspondences

#### Visualization:

- Classical Multidimensional Scaling (using bundled [Spectra](#) for eigendecompositions)
- A powerful, extensible, and easy to use 3D visualizer



### 2.1 Dependencies

- [Eigen](#) (version 3.3 or newer) [**required**]
- [Pangolin](#) (built with Eigen enabled) [**optional**; needed for visualization modules and most examples]

### 2.2 How to build

**cilantro** is developed and tested on Ubuntu variants (18.04 and newer) using [CMake](#). To clone and build the library (with bundled examples), execute the following in a terminal:

```
git clone https://github.com/kzampog/cilantro.git
cd cilantro
mkdir build
cd build
cmake ..
make -j
```