

代码文件结构

- `scan` (`sensor_msgs/LaserScan`)
支持二维和三维(例如, 使用轴向旋转的平面激光扫描仪). 如果`num_laser_scans`在`Configuration`配置为1, 那么这个topic 将被用SLAM的输入. 如果`num_laser_scans`大于1, 那么多个被编号的scan topics (比如`scan1`、`scan2`、`scan3`、.....直到并包括`num_laser_scans`)将被用作SLAM的输入.
- `points2` (`sensor_msgs/PointCloud2`)
如果 `num_point_clouds`在`Configuration`配置为1, 那么这个topic 将被用SLAM的输入。如果`num_point_clouds`大于1, 那么多个被编号的`points2` topics (比如`points2_1`、`points2_2`、`points2_3`、.....直到并包括`num_point_clouds`)将被用作SLAM的输入.
- `imu` (`sensor_msgs/Imu`)
支持2D(可选)和3D(必需)。这个topic 将被用作SLAM的输入.
- `odom` (`nav_msgs/Odometry`)
支持2D(可选)和3D(可选)。如果在`Configuration`中使能了`use_odometry`, 那么这个topic将被用作SLAM的输入.

测试分析

reference

[cartograph 安装测试](#)

[cartograph 前端优化](#)

[cartograph 论文阅读](#)

[cartograph 源码阅读](#)

[cartograph 占据概率推导](#)

[cartographer理论及实现浅析](#)

[cartograph 前后端总结](#)

[cartograph-ros doc](#)

[cartograph-dataset](#)

[cartographer-3D SLAM](#)

[cartographer + velodyne64 + IMU建图](#)

[Cartographer Laser-SLAM 算法流程 -- 调试 Cartographer](#)