

# EG—SLAM

---

## 资料：

---

```
1 | https://blog.csdn.net/tiancailx/category\_10582260.html
2 |
3 | https://blog.csdn.net/tiancailx/article/details/110822624
4 |
5 | https://github.com/xiangli0608/Creating-2D-laser-slam-from-scratch
```

## 目的：

---

1. 深入学习SLAM
2. 实际编写SLAM，提高代码工程能力

## 简介：

---

从零二维激光SLAM 算法

## 算法流程：

---

前端：

前端，通过PCL-ICP算法，进行帧间匹配求得相对位姿变换

后端：

激光SLAM中，后端进行优化和建图，主流的SLAM框架会单独分一个线程来执行后端，因为不管是建图还是优化，都需要占用相当的计算资源，为了保证实时性，

采取多线程，限制后端优化和建图的规模，也有很多小tricks，比如建立关键帧采用滑动窗口，丢掉观测只优化位姿图等'

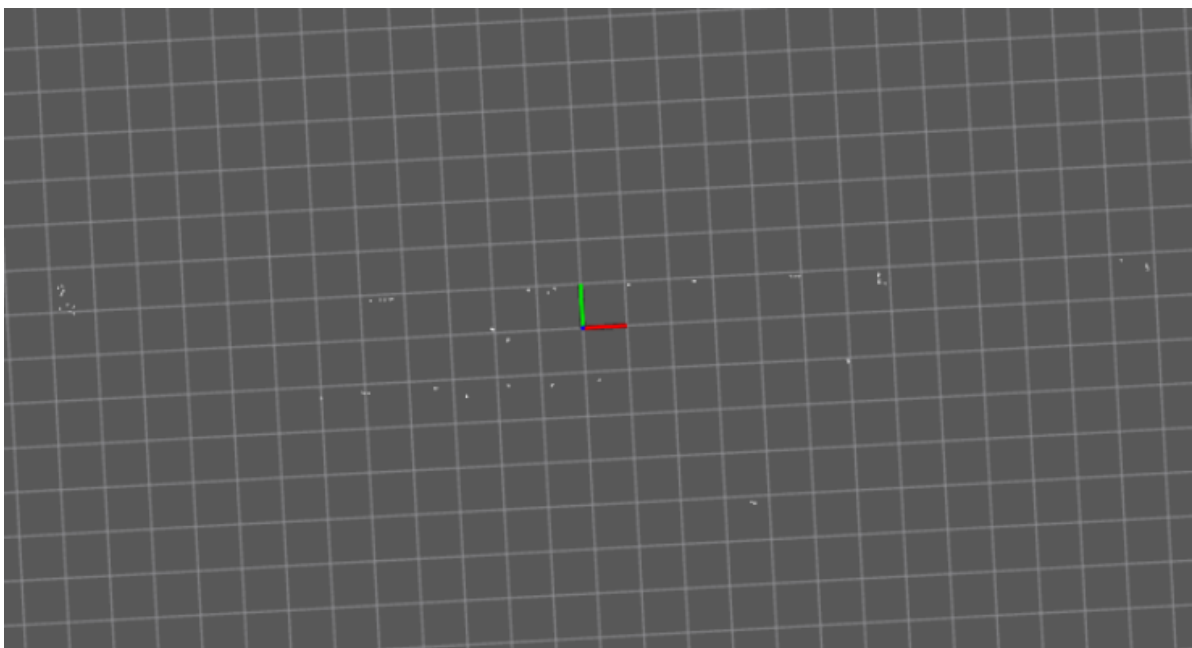
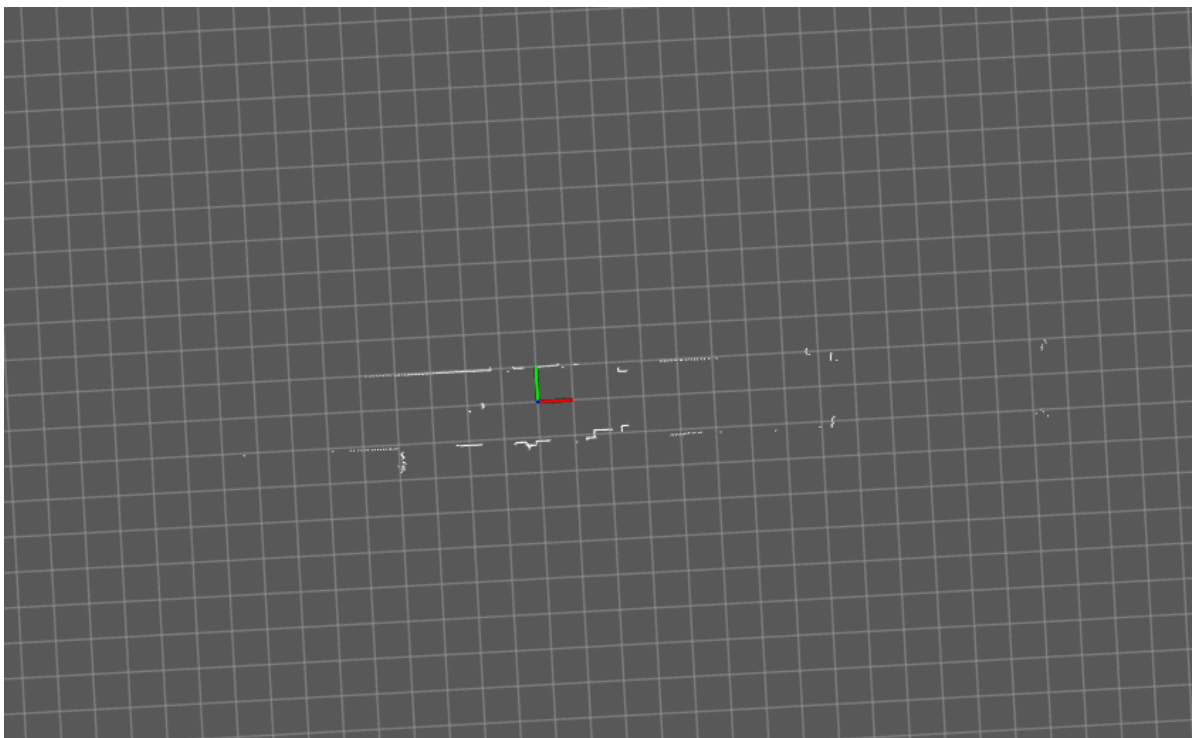
建图：本位采用了经典的占据栅格地图，其本质就是连续的地图离散化，变成一个一个栅格，如果被激光hit，则设计那个格子为占据，否则则设定为空闲，没有被激光扫到的格子状态为未知，如果用三个值来表示的化，占据的就是100，空闲的是0，未知的是-1，随后我们维护一个数组，把每个格子的状态放进去，在建图的时候把这个一维数组进行二维展开，再根据每个格子的状态用颜色进行区分（系统自动完成），就完成了了一个最简单的占据栅格地图。

回环

前端进行扫描匹配，后端进行优化，回环消除累计误差

## 1. 提取关键帧

---



## 2. 使用PCL 进行雷达消息类型转换

sensor\_msgs/LaserScan转换成的sensor\_msgs/PointCloud2

## 3. ICP 帧间匹配

使用转换格式后的点云，调用ICP算法进行相邻两帧雷达数据间坐标变换的计算

```
transfrom: (0.00671138, 0.00275577, 0.123683)
ICP计算用时: 0.0426161 秒。

转换数据格式用时: 5.8385e-05 秒。
transfrom: (0.00561491, -0.00599963, 0.456653)
ICP计算用时: 0.0493912 秒。

转换数据格式用时: 5.6683e-05 秒。
transfrom: (0.00300559, 0.00259367, -0.00352352)
ICP计算用时: 0.0394016 秒。

转换数据格式用时: 6.0527e-05 秒。
transfrom: (0.00510993, 0.0269651, -0.19295)
ICP计算用时: 0.0402743 秒。

转换数据格式用时: 7.0754e-05 秒。
transfrom: (0.00603233, 0.0114045, -0.15819)
ICP计算用时: 0.0409717 秒。

转换数据格式用时: 0.00018613 秒。
transfrom: (-0.00203995, 0.00506146, 0.351611)
ICP计算用时: 0.0525758 秒。
```

## 4. PL-ICP的帧间匹配

ICP的计算时间以及精度都不太好，使用ICP改进算法 PL-ICL算法来计算相邻帧间的坐标变换PL

PL-ICP 使用点到线距离最小的方式进行ICP的计算，收敛速度快，精度高

PL-ICP 作者开源资料：

- 1 <https://censi.science/software/csm/>.
- 2 wiki 地址:
- 3 [http://wiki.ros.org/scan\\_tools?distro=kinetic](http://wiki.ros.org/scan_tools?distro=kinetic)

功能包介绍：

- laser\_ortho\_projector: 将切斜的雷达数据投影到平面上.
- laser\_scan\_matcher: 基于pl-icp的扫描匹配的实现，并进行了位姿累加
- laser\_scan\_sparsifier: 对雷达数据进行稀疏处理
- laser\_scan\_splitter: 将一帧雷达数据分段，并发布出去
- ncd\_parser: 读取New College Dataset，转换成ros的scan 与 odometry 发布出去
- polar\_scan\_matcher: 基于Polar Scan Matcher的扫描匹配器的ros实现

- scan\_to\_cloud\_converter: 将 sensor\_msgs/LaserScan 数据转成 sensor\_msgs/PointCloud2 的数据格式

```
转换数据格式用时: 6.3383e-05 秒。  
transform: (0.000239358, -5.07149e-05, 0.000308067)  
PLICP计算用时: 0.00157147 秒。  
  
转换数据格式用时: 9.675e-05 秒。  
transform: (-6.42462e-05, -0.000387982, -0.00231315)  
PLICP计算用时: 0.00254317 秒。  
  
转换数据格式用时: 0.000116659 秒。  
transform: (-0.000175684, 0.000321544, 0.000859921)  
PLICP计算用时: 0.00277954 秒。  
  
转换数据格式用时: 0.000109408 秒。  
transform: (-4.13895e-05, 6.42636e-05, 0.00822606)  
PLICP计算用时: 0.0028693 秒。
```

##

