深蓝学院
shenlanxueyuan.com

多传感器融合定位

第二章作业思路提示

主讲人 陈梓杰

纲要

深蓝学院
shenlanxueyuan.com

- 代码运行问题
- ICP_SVD补全思路
- ICP_SVD补全注意事项
- 学习建议

代码运行问题

深蓝学院 shenlanxueyuan.com

- 要想成功运行代码，必须先修改/workspace/assignments/02-lidar-odometry basic/src/lidar_localization/config/front_end/config.yaml
- 不然会提示“boost”或“tf为nan”错误

ICP_SVD补全思路

深蓝学院 shenlanxueyuan.com

4 Rigid motion computation – summary

Let us summarize the steps to computing the optimal translation $\mathbf{t}$ and rotation R that minimize

$$\sum_{i=1}^{n} w_i \left\| (R\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i \right\|^2 .$$

1. Compute the weighted centroids of both point sets:

$$\bar{\mathbf{p}} = \frac{\sum_{i=1}^{n} w_i \mathbf{p}_i}{\sum_{i=1}^{n} w_i}, \quad \bar{\mathbf{q}} = \frac{\sum_{i=1}^{n} w_i \mathbf{q}_i}{\sum_{i=1}^{n} w_i} .$$

2. Compute the centered vectors

$$\mathbf{x}_i := \mathbf{p}_i - \bar{\mathbf{p}}, \quad \mathbf{y}_i := \mathbf{q}_i - \bar{\mathbf{q}}, \qquad i = 1, 2, \ldots, n.$$

3. Compute the $d \times d$ covariance matrix

$$S = XWY^{\mathsf{T}},$$

where X and Y are the $d \times n$ matrices that have $\mathbf{x}_i$ and $\mathbf{y}_i$ as their columns, respectively, and $W = \mathrm{diag}(w_1, w_2, \ldots, w_n)$.

4. Compute the singular value decomposition $S = U\Sigma V^{\mathsf{T}}$. The rotation we are looking for is then

$$R = V \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & \det(VU^{\mathsf{T}}) \end{pmatrix} U^{\mathsf{T}}.$$

5. Compute the optimal translation as

$$\mathbf{t} = \bar{\mathbf{q}} - R\bar{\mathbf{p}}.$$

左图摘自论文《Least-Squares Rigid Motion Using SVD》

ICP_SVD补全注意事项

深蓝学院 shenlanxueyuan.com

- PPT中，**x是目标点云，y是源点云**；代码中，**x是源点云，y是目标点云**。切记不要弄混
- 建议在GetTrasnform()函数中加入旋转矩阵是否为“reflection”的判断。具体可以参考：《Least-Squares Rigid Motion Using SVD》、《Least-Squares Fitting of Two 3-D Point Sets》
- ScanMatch()中，每次计算完都要对旋转矩阵进一步处理，保证其满足旋转矩阵的性质，不然会出现**尺度漂移很大的问题**。处理方式有：

方法一：

```
Eigen::Quaternionf q(result_pose.block<3,3>(0,0);
q.normalize();
result_pose.block<3,3>(0,0) = q.toRotationMatrix();
```

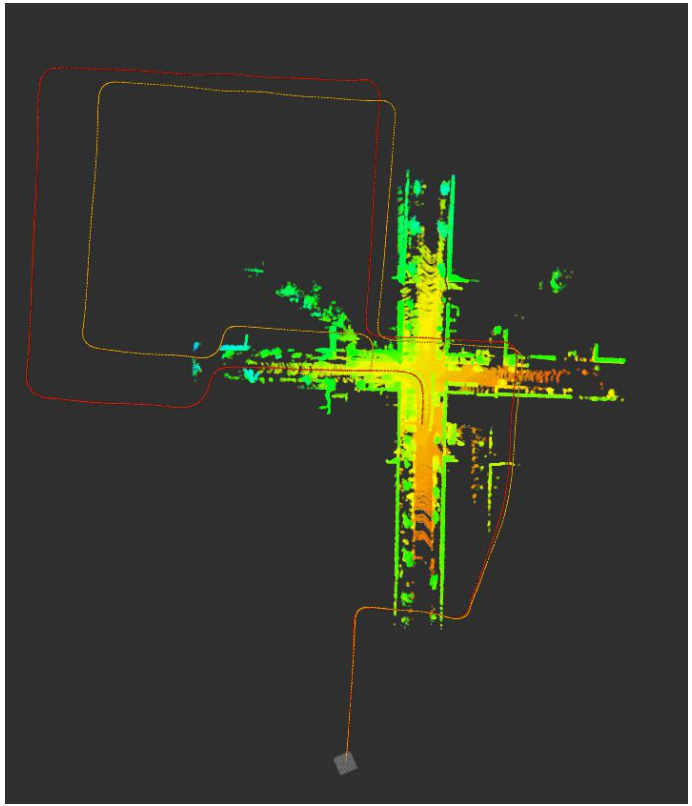
直接对四元数做归一化处理，最简洁有效

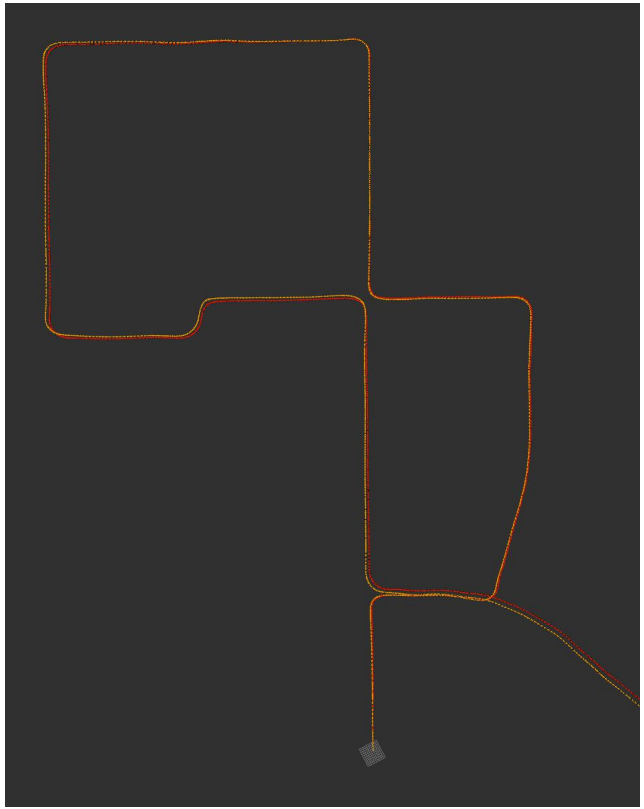
方法二：

```
Eigen::JacobiSVD<Eigen::MatrixXf> svd(
    result_pose.block<3,3> (0,0),
    Eigen::ComputeFullU | Eigen::ComputeFullV);
result_pose.block<3,3>(0,0)
    = svd.matrixU() * svd.matrixV().transpose();
```

参考自《机器人学中的状态估计》
公式7.213上面，没有编号那条公式

ICP_SVD补全注意事项





- 左图为某位学员的代码效果
- 右图是我为它添加四元数归一化的效果
- 效果提升是非常明显的

学习建议

- **尝试修改yaml文件中的参数，体会不同参数对里程计精度和耗时的影响：**

1. Voxel Grid为什么要设置成0.6？调大点或小一点会有什么变化？

2. ICP的max_corr_dist调大点会提升性能？

3. NDT的res调小会影响里程计的效率嘛？

- **仿照课程例程的框架，自己实现其他配准方法，借助课程提供的代码框架，体会目前开源配准方法的优缺点**

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening