

深蓝学院激光 SLAM 第三期第三次作业

一·本次作业练习目标

巩固课程所学知识，通过代码实践利用里程计完成激光雷达的运动畸变去除，加深对激光雷达数学模型与 ICP 算法的理解，为下一节课重点拓展 ICP 算法学习做铺垫。

二·作业计分原则

满分为 12 分，10 分为优秀，8 分为良好，6 分为及格。

三·作业提交说明

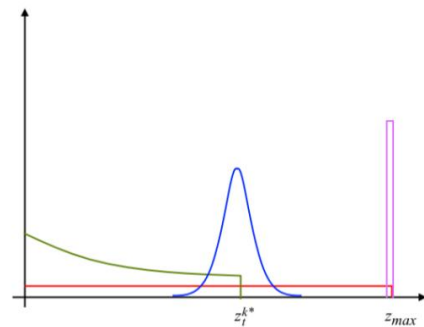
需提供完整的 PDF 报告及代码。公式推导题可以手写照相粘进 PDF，也可以直接写进 PDF。

四·作业题目说明

1. 补充去除激光雷达运动畸变模块的代码；(6 分)
2. 阅读论文 Least-Squares Fitting of Two 3-D Points Sets，推导并证明已知对应点的 ICP 求解方法；(2 分)
3. 阅读论文 Precise indoor localization for mobile laser scanner 前两章，回答问题。(2 分)
 - (1) 根据第二章内容，简述激光雷达测距原理；
 - (2) 简要介绍一下右图的含义。

4. 简答题，开放性答案：设计使用 IMU 去除激光雷达运动畸变的方法并回答问题。(2 分)

- (1) 仅用 IMU 去除运动畸变可能会有哪些不足之处？
- (2) 在仅有 IMU 和激光雷达传感器的情况下，你会如何设计运动畸变去除方案(平移+旋转)，达到较好的畸变去除效果？



五·作业提示与学习材料

第一题代码说明：

本题目为实现一个里程计去除激光雷达运动畸变的代码模块，作业里面有两个工程：champion_nav_msgs 和 LaserUndistortion。大家需要首先编译安装 champion_nav_msgs，按照 champion_nav_msgs 的 readme 文件执行即可，或运行命令 `sudo bash install.sh`，注意如果你的 ubuntu 版本不是 kinetic，要将所有 kinetic 的地方修改成你的 ros 版本。

程序运行过程为：

Step1：实现 207 行 LidarMotionCalibration 函数，并用 `catkin_make` 命令进行编译；

Step2：在 LaserUndistortion 下，进行 `source`：`source devel/setup.bash`；

Step3：运行 launch 文件：`roslaunch LaserUndistortion LaserUndistortion.launch`，执行本条指令

的时候，必须保证没有任何 ROS 节点在运行，roscore 也要关闭；

Step4：进入到 /bag 目录下,运行指令:rosbag play --clock laser.bag;

Step5：如果一切正常,则会看到 pcl 的可视化界面,当可视化界面中存在数据的时候,按 R 键即可看到结果（红色为畸变矫正前,绿色为畸变矫正后）。

Least-Squares Fitting of Two 3-D Points Sets 这篇文章虽然比较老，但对 ICP 的推导非常明了详细，ICP 是激光 slam 的基础，请一定要花时间把原理弄清。

Precise indoor localization for mobile laserscanner 2015 这篇硕士论文比较详细的介绍了激光 slam 从底层架构到具体实现，并比较全面的测试对比了 GMapping，Hector Slam，Karto 三种激光 slam 算法。可以作为不错的激光 slam 入门材料，推荐大家有兴趣可以完整的阅读一遍。

有关于 tf 库对位姿（平移和旋转）的插值函数：

```
TFSIMD_FORCE_INLINE Vector3 tf::Vector3::lerp ( const Vector3 & v,  
                                                const tfScalar & t  
                                                ) const [inline]
```

Return the linear interpolation between this and another vector.

Parameters:

- v The other vector
- t The ration of this to v (t = 0 => return this, t=1 => return other)

Definition at line 233 of file Vector3.h.

```
Quaternion tf::Quaternion::slerp ( const Quaternion & q,  
                                   const tfScalar & t  
                                   ) const [inline]
```

Return the quaternion which is the result of Spherical Linear Interpolation between this and the other quaternion.

Parameters:

- q The other quaternion to interpolate with
- t The ratio between this and q to interpolate. If t = 0 the result is this, if t=1 the result is q. Slerp interpolates assuming constant velocity.

Definition at line 314 of file Quaternion.h.

http://docs.ros.org/jade/api/tf/html/c++/classtf_1_1Vector3.html#a49220836c2fc359bf9e7feb307cade5a

http://docs.ros.org/jade/api/tf/html/c++/classtf_1_1Quaternion.html#affa098b16b0091af8b71bf533b5494a

同样 Eigen 库也有类似的接口函数可以进行插值，感兴趣可以自己上网查找学习。

学习 tf 库的材料：

<http://www.guyuehome.com/279>

<http://www.guyuehome.com/355>

想了解更多有关 tf 库与坐标变化的内容可以在 ROS 官网进行学习，部分接口内容可以现用现查。

<http://wiki.ros.org/tf/Tutorials>