

智能移动机器人期中报告

唐治江-2024E8021682292-智能学院-人工智能

Code: <https://github.com/ZhijiangTang/Intelligent-Mobile-Robot-Midterm-Report>

作业1

目标: 理解运动传感器和基于航位推算的机器人运动估计方法

1.1 运动不确定性模型

给定控制量 $u = (\delta_{rot}, \delta_{trans})$ ，当前位姿 $x = (x, y, \theta)$ ，噪声参数 $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$

- 噪声采样:

$$\begin{cases} \hat{\delta}_{rot} = \delta_{rot} + \epsilon_{rot}, & \epsilon_{rot} \sim \mathcal{N}(0, \alpha_1 |\delta_{rot}| + \alpha_2 |\delta_{trans}|) \\ \hat{\delta}_{trans} = \delta_{trans} + \epsilon_{trans}, & \epsilon_{trans} \sim \mathcal{N}(0, \alpha_3 |\delta_{trans}| + \alpha_4 |\delta_{rot}|) \end{cases}$$

- 位姿更新:

$$\begin{cases} x' = x + \hat{\delta}_{trans} \cdot \cos\left(\theta + \frac{\hat{\delta}_{rot}}{2}\right) \\ y' = y + \hat{\delta}_{trans} \cdot \sin\left(\theta + \frac{\hat{\delta}_{rot}}{2}\right) \\ \theta' = \theta + \hat{\delta}_{rot} \end{cases}$$

- 设 IMU 测量序列为 $\{\psi_k\}$ ，时间对齐后，相对旋转量为 $\delta_{rot}^{(k)} = \psi_k - \psi_{k-1}$ ，控制量构成为 $u_k = (\delta_{rot}^{(k)}, d_k)$

- 系统状态递推方程:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, u_k, \alpha) = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \hat{\delta}_{trans}^{(k)} \cos\left(\theta_{k-1} + \frac{\hat{\delta}_{rot}^{(k)}}{2}\right) \\ \hat{\delta}_{trans}^{(k)} \sin\left(\theta_{k-1} + \frac{\hat{\delta}_{rot}^{(k)}}{2}\right) \\ \hat{\delta}_{rot}^{(k)} \end{bmatrix}$$

其中噪声项满足:
$$\begin{cases} \hat{\delta}_{rot}^{(k)} = \delta_{rot}^{(k)} + \epsilon_{rot}^{(k)} \\ \hat{\delta}_{trans}^{(k)} = \delta_{trans}^{(k)} + \epsilon_{trans}^{(k)} \end{cases}$$

1.2 结果

误差持续累加。利用运动不确定性模型建立的机器人轨迹图如图1所示，从图中可以看出，随着机器人步数的增加，误差也在持续累加，导致越来越偏离轨道。图2为将第一圈轨迹绘制在真实地图中，从图中也可以看出，前面半程轨迹还比较正确，后面半程机器人轨迹就开始偏移，导致最终的轨迹都走进了墙体里面。

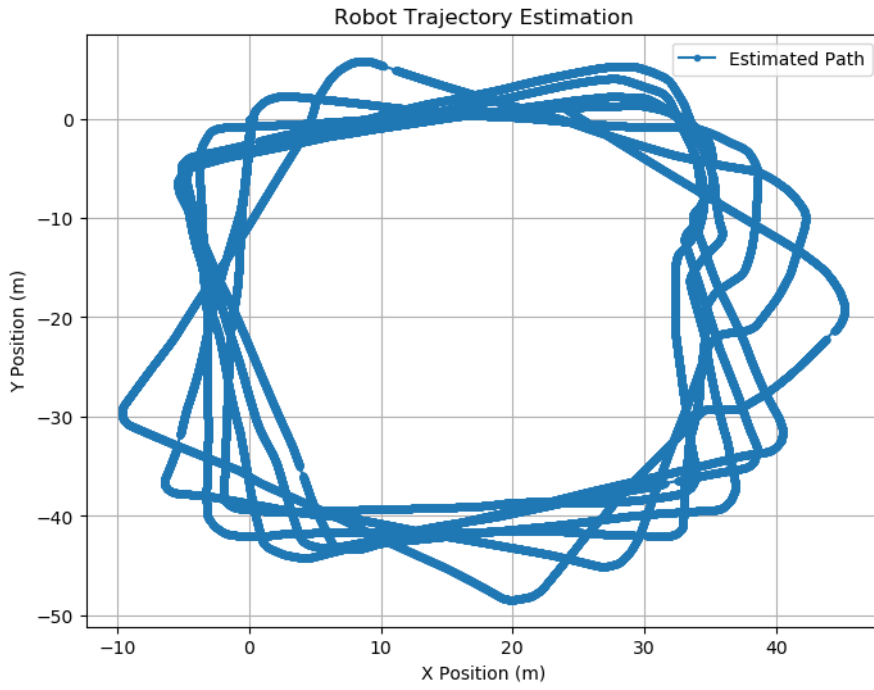


图1：利用运动不确定性模型建立的机器人轨迹图



图2：第一圈轨迹图与真实地图

作业2

目标：理解激光雷达数据

2.1 激光雷达坐标转换数学模型

设 t 为时间戳索引， $t \in \{0, 1, \dots, T-1\}$ ， $\mathbf{p}_t = [t_x, t_y, \theta_t]^T$ 表示时刻 t 的位姿（平移分量和旋转角）， $r_i^{(t)}$ 为时刻 t 第 i 个激光点的距离测量值， $i \in \{0, 1, \dots, 359\}$ ， $\Delta\phi$ 为角度分辨率参数， s 为单位换算系数。

- 极坐标转笛卡尔坐标。对每个激光点进行极坐标到局部坐标系的转换：

$$\begin{cases} x_i^{(t)} = \frac{r_i^{(t)}}{s} \cdot \cos(\phi_i) \\ y_i^{(t)} = \frac{r_i^{(t)}}{s} \cdot \sin(\phi_i) \end{cases}$$

其中扫描角度计算为： $\phi_i = i \cdot \Delta\phi$

坐标系旋转。定义旋转矩阵 $\mathbf{R}(\theta_t) = \begin{bmatrix} \cos \theta_t & -\sin \theta_t \\ \sin \theta_t & \cos \theta_t \end{bmatrix}$ ，对每个点进行旋转变换：

$$\begin{bmatrix} \tilde{x}_i^{(t)} \\ \tilde{y}_i^{(t)} \end{bmatrix} = \mathbf{R}(\theta_t) \begin{bmatrix} x_i^{(t)} \\ y_i^{(t)} \end{bmatrix}$$

整体变换公式。应用位姿的平移分量 $\begin{cases} X_i^{(t)} = \tilde{x}_i^{(t)} + t_x \\ Y_i^{(t)} = \tilde{y}_i^{(t)} + t_y \end{cases}$ ，世界坐标系坐标为：

$$\begin{bmatrix} X_i^{(t)} \\ Y_i^{(t)} \end{bmatrix} = \mathbf{R}(\theta_t) \begin{bmatrix} \frac{r_i^{(t)}}{s} \cos(i\Delta\phi) \\ \frac{r_i^{(t)}}{s} \sin(i\Delta\phi) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

2.2 简易投票法

1. 创建栅格地图：

- 根据栅格分辨率 res 和地图范围 (wid, len) 初始化二维数组 grid，初值为0

2. 投影激光点：

对于激光点 (px, py)：

- 将其变换到全局坐标 (gx, gy)
- 根据 res 计算对应的栅格索引 (ix, iy)
- 若在地图范围内，则 grid[ix][iy] += 1

3. 阈值判断：

对 grid 中的每个栅格：

- 若计数值 > threshold，则认为该格为障碍物
- 否则为空或未知

4. 输出结果：

- 得到可视化的栅格地图或障碍物分布

2.3 结果

雷达点云图如图3所示。从图中可以看出，中央和两侧有较为密集且轮廓平直的点云，从图中可以推测，雷达在环境中多次经过同一条走廊或区域，导致同一位置在点云中被重复扫描，点云在那些区域更加密集。这是图2的真是地图是一致的。

投票栅格图如图4所示。设置分辨率res为0.2m，阈值threshold为10。在0.2m的分辨率下可以得到一个很好的栅格图，其与图2的真实地图非常相似，而其中的可移动区域也比较明显。

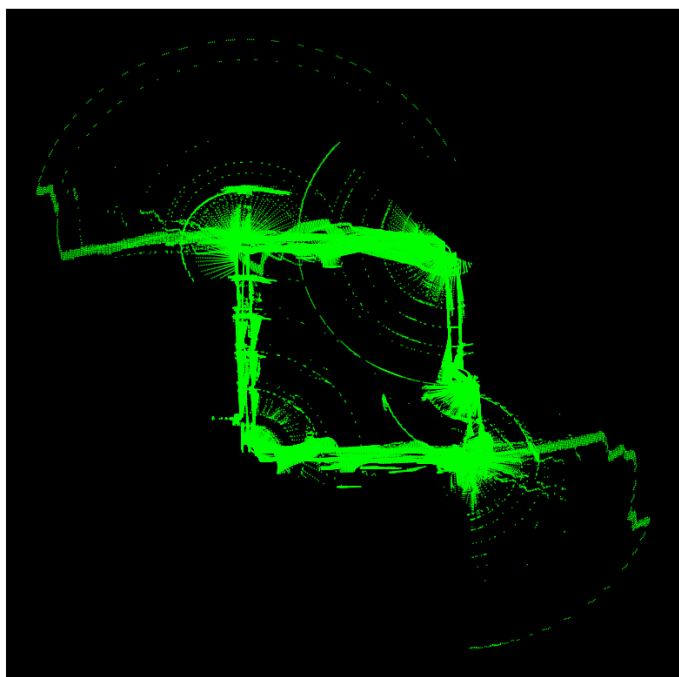


图3：雷达点云图

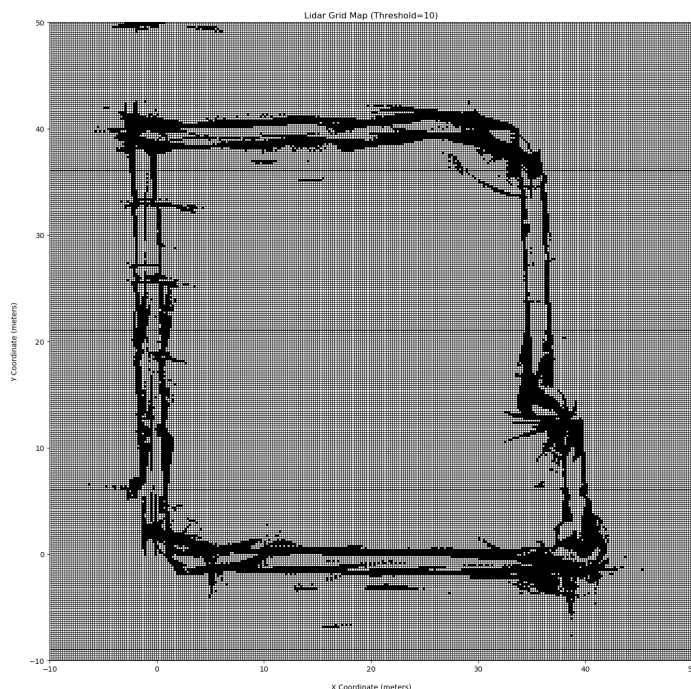


图4：投票栅格图

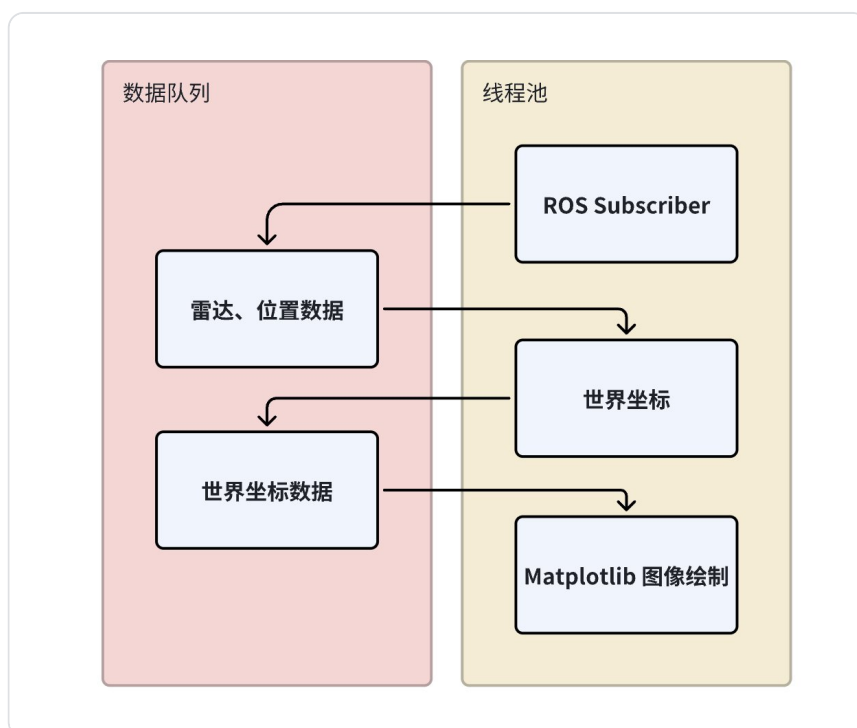
作业3

目标：学习ROS2 foxy版本

3.1 多线程

使用ROS2重写作业2代码时，往往有较高的时间复杂度，如果单线程的话会阻塞雷达、位置信号的采集。所以基于ROS2重写作业2的同时，也将其改为多线程。

多线程示例如图5所示，总共包含三个线程，分别处理：获取雷达位置信号、将信号转化为世界坐标并投票、绘制图像。同时包含两个数据队列：雷达位置数据和世界坐标数据。



3.2 结果

实时栅格地图生成录制有[示例视频](#)。最终生成的最终生成的雷达点云图和栅格投票图如图6所示，其结果与作业2类似，能很好地建模障碍，与图2的真实地图非常相似。

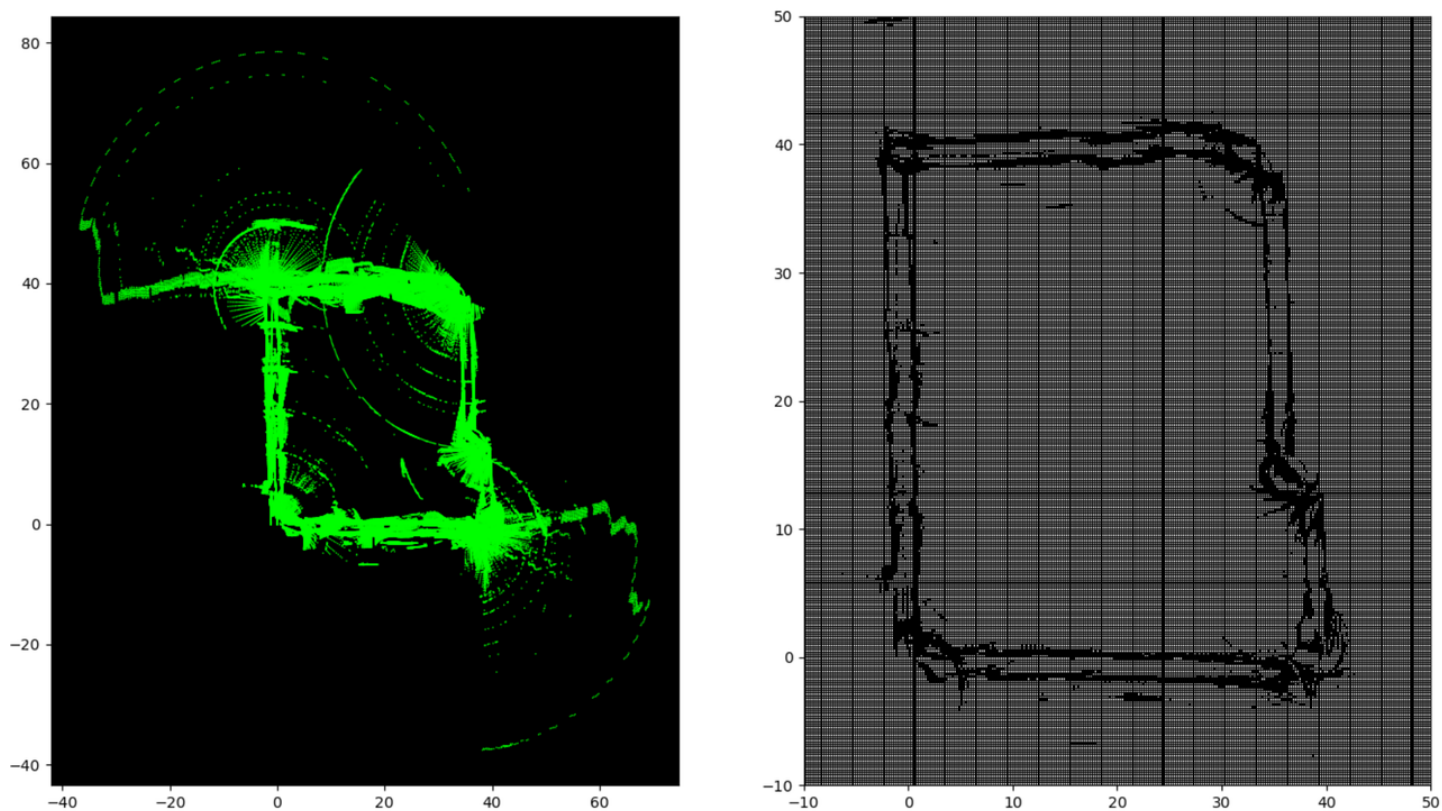


图6：最终生成的雷达点云图（左）和栅格投票图（右）