张 文泰

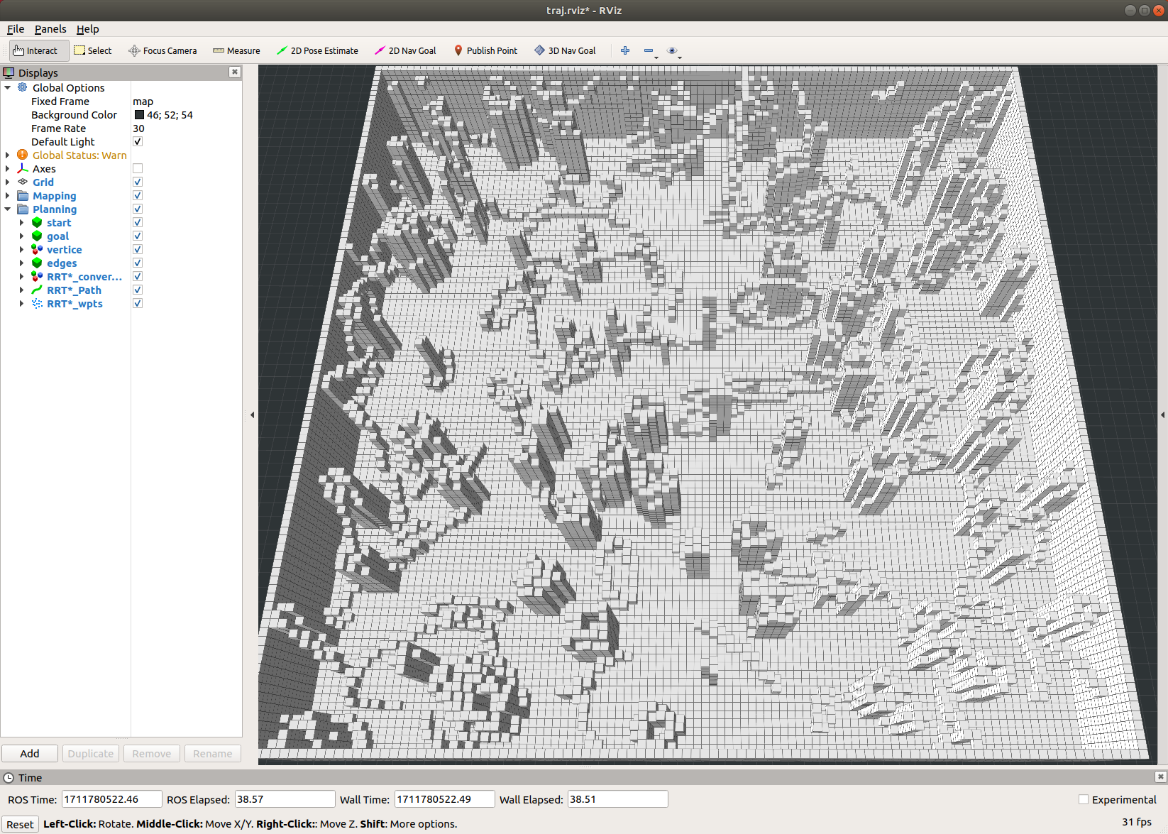
学号：21009101463

**机器人路径规划**

第三章作业

实验一 ROS版本作业

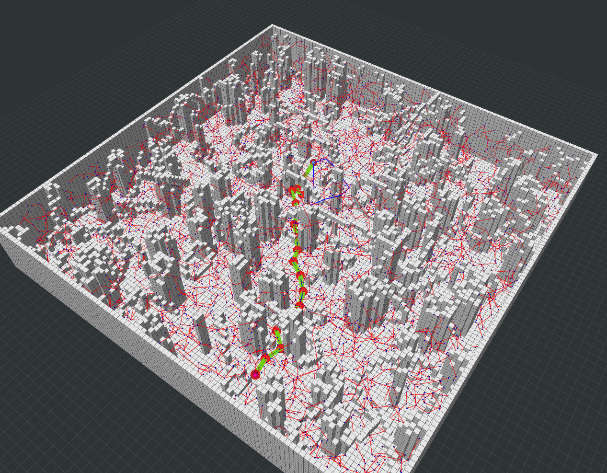
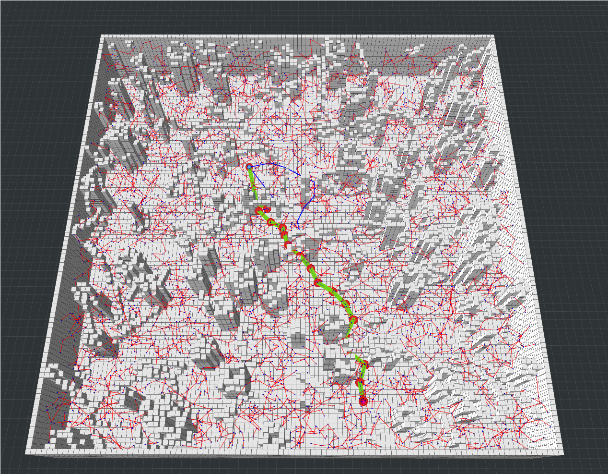
1. 实验过程



ROS版本作业过程图

1. 实验结果

路径规划结果如图所示：



实验二 MATLAB版本作业

1. 代码补充与分析



主程序代码

这段代码的整体逻辑如下：

1. 首先，使用一个循环来执行3000次迭代，在每次迭代中，从一个随机点x\_rand开始。

2. 然后，找到距离x\_rand最近的点x\_near，并在图上以红色圆点表示。

3. 接下来，使用Steer函数从x\_near移动到一个新的点x\_new，如果从 x\_near到x\_new的路径没有碰撞（即collisionChecking返回true），则继续下一次迭代，否则就更新计数器count，并将x\_new的坐标、x\_near的坐标、距离以及索引信息存储在数据结构T.v中。

4. 如果x\_new到目标点goal的距离小于阈值Thr，则跳出循环。

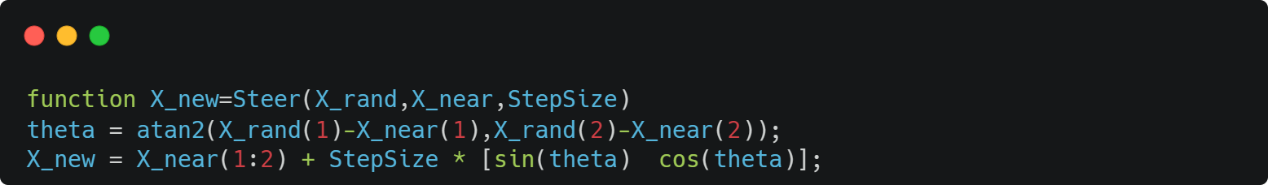
5. 最后在图上绘制从x\_near到x\_new的连线，并暂停一段时间使得RRT扩展过程容易观察。

除了主程序之外，还有几个函数代码需要分析：



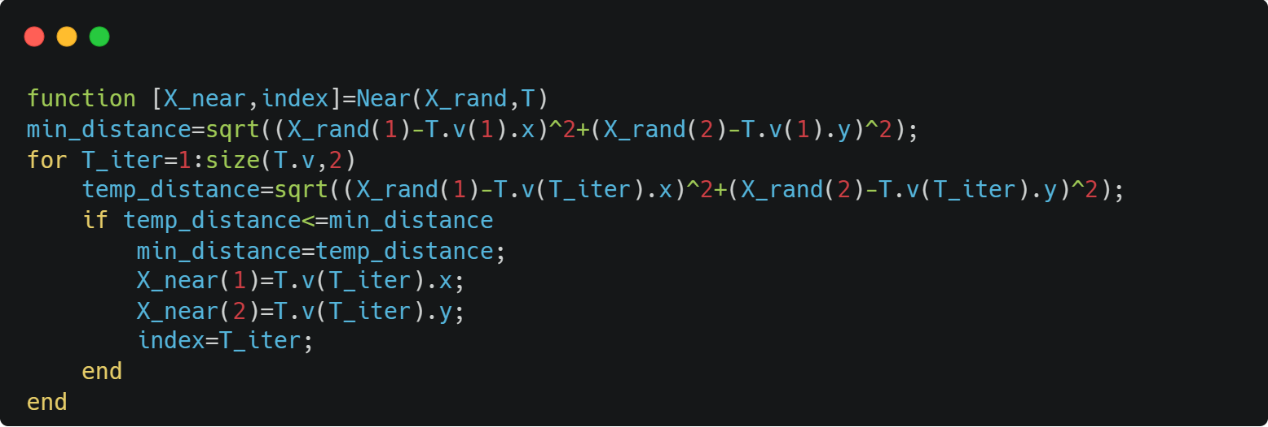
Sample函数代码

Sample函数的目的是在路径规划中生成一个随机采样点，用于后续的路径搜索和规划。基本逻辑为，首先生成一个随机点X\_rand，如果随机数小于0.5，会在地图的范围内随机选择一个点，X\_rand(1)是一个在0到地图的行数之间均匀分布的随机数乘以地图的行数，X\_rand(2)是一个在0到地图的列数之间均匀分布的随机数乘以地图的列数。否则如果随机数大于等于0.5，直接将 X\_rand设置为目标点goal。



Steer函数代码

Steer函数的目的是在从一个点X\_near移动到一个新的点X\_new，以便生成路径，其思路为，首先使用atan2函数计算从X\_near到X\_rand的方向角theta，接下来，根据StepSize（步长）和theta计算新的点X\_new：X\_new的 x 坐标是X\_near的 x 坐标加上 StepSize \* sin(theta)，X\_new的 y 坐标是X\_near的 y 坐标加上StepSize \* cos(theta)。



Near函数代码

Near函数用于在路径规划中找到距离随机点X\_rand最近的点，其逻辑为：

1. 初始化最小距离：计算随机点 X\_rand到图中第一个点T.v(1)的距离，并将其作为初始最小距离。

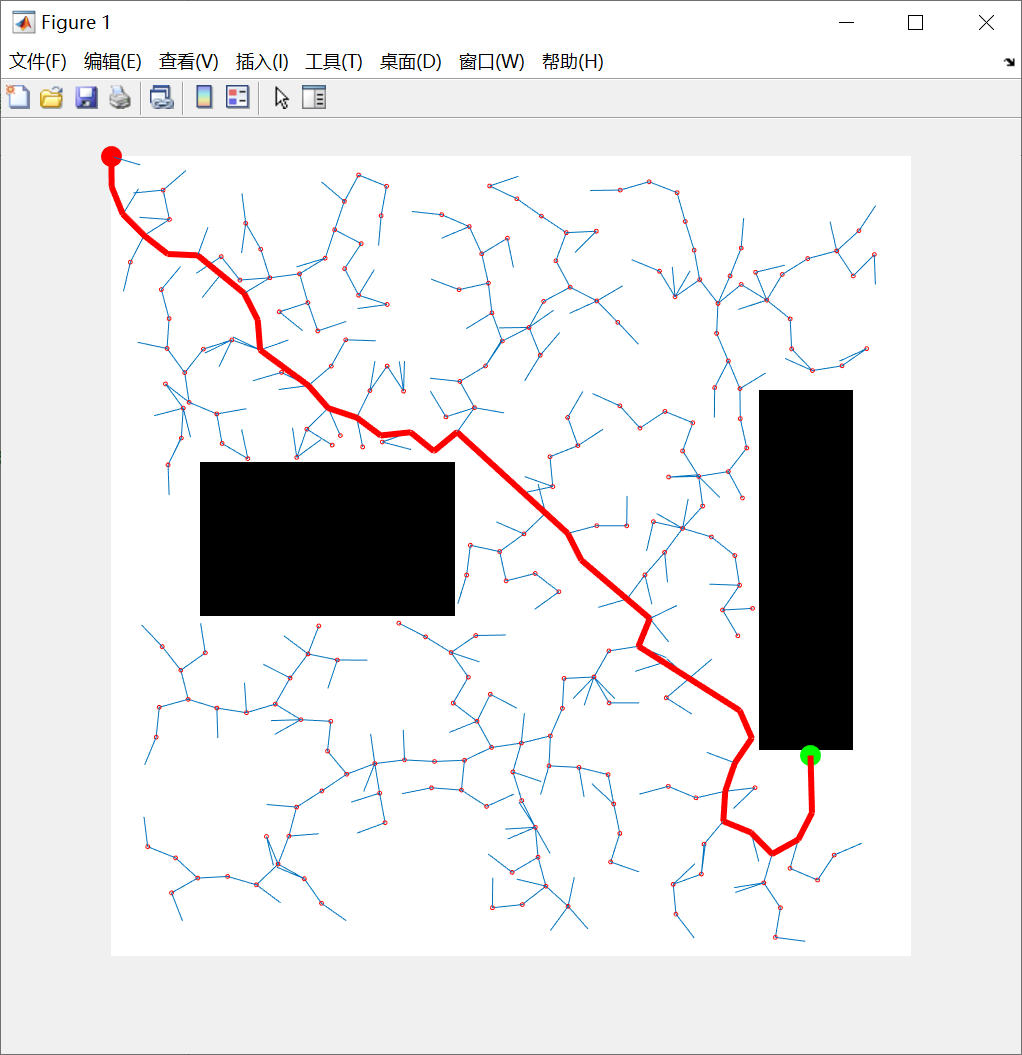
2. 遍历：遍历图中的所有点（通过T.v访问）。

3. 计算临时距离：对于每个点，计算X\_rand到该点的距离，并将其存储在 temp\_distance中。

4. 更新最小距离和最近点：如果temp\_distance小于等于当前最小距离，更新最小距离，并将该点的坐标存储在X\_near中，同时记录该点的索引。

5. 返回结果：最终，函数返回最近的点X\_near和其索引。

1. 运行结果与感受



MATLAB版本作业运行结果

代码完成后运行多次，结果均正确，成功完成了RRT路径规划，本次实验也加深了我对RRT算法的认识与理解，锻炼了我的MATLAB代码能力与算法逻辑能力，让我对机器人路径规划课程知识有了新的感悟。