

# matlab

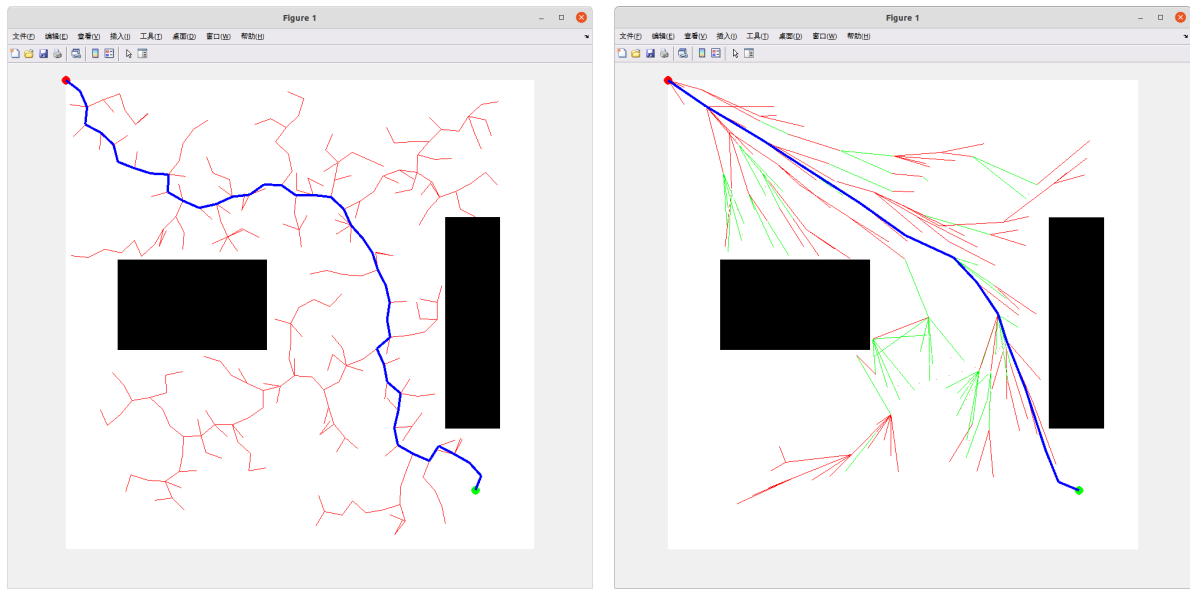


图1 matlab下RRT和RRT\*仿真效果图

在matlab下实现了RRT和RRT\*算法，分别如上面左图和右图所示。

# ros

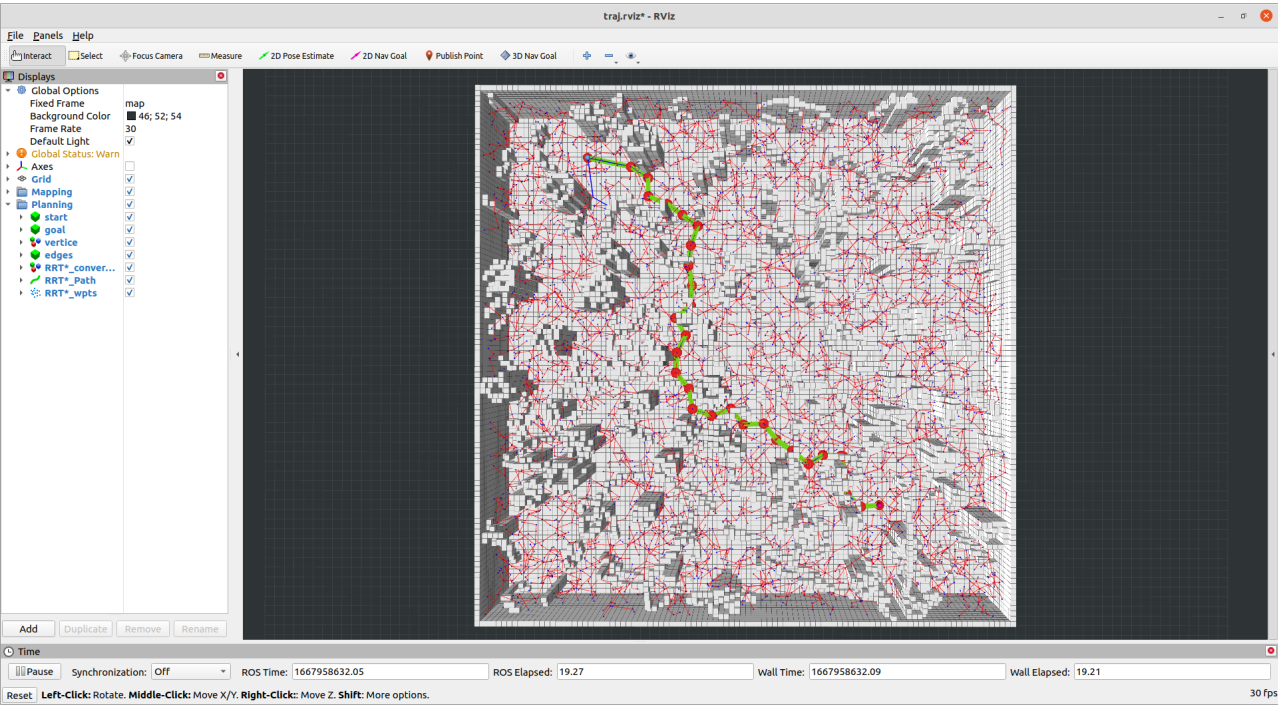


图2 RRT路径搜索效果图

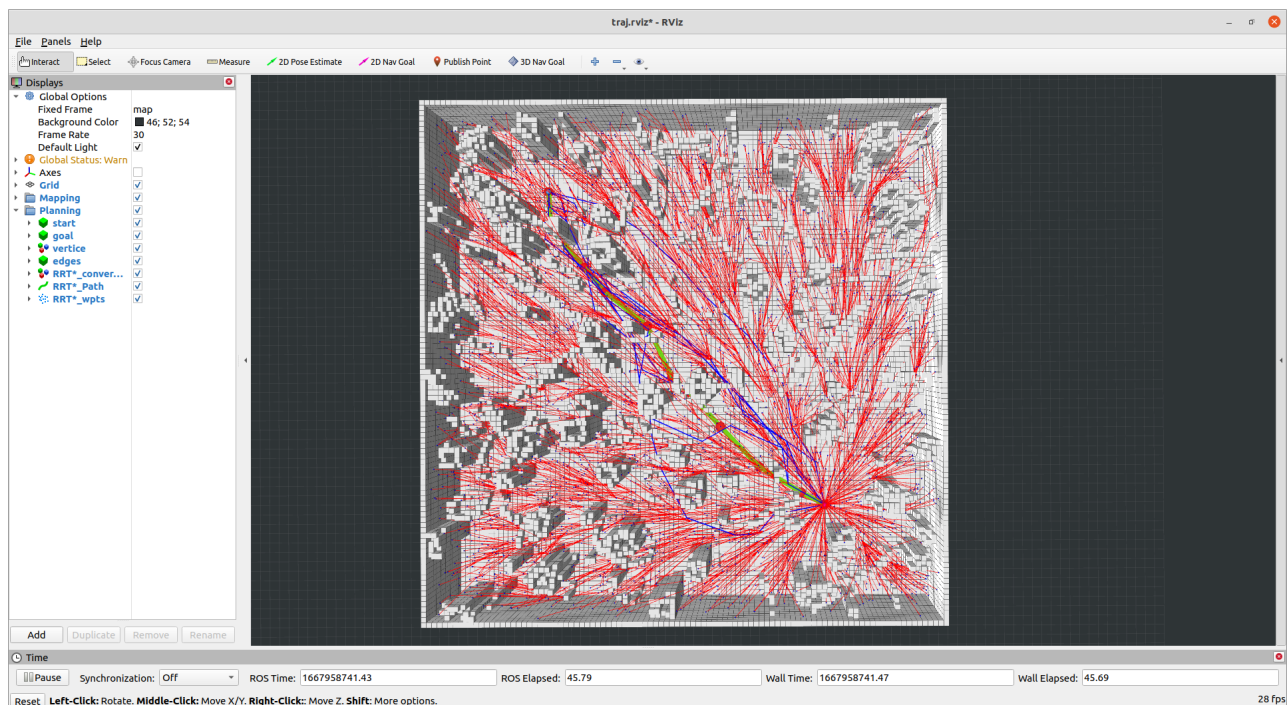


图3 RRT\*路径搜索效果图

在ROS下实现的RRT和RRT\*算法，分别如上面左图和右图所示。

## 结果分析

从matlab和ros下的仿真结果可以看出，RRT\*搜索的路径相对与RRT更优，具体表现为路径连续性更高更加平滑、路径更短，这是因为RRT的搜索策略是完全随机的，RRT\*在RRT的基础上加入了**对父节点选择（chooseParent）和重新布线（rewire）**的操作。

chooseParent操作保证在每次随机搜索一个新节点后，起点到新节点的路径是最短的；rewire操作是对新节点周围一定半径内节点的路径进行更新，保证扩展新节点后周围节点的路径最优。因此，RRT\*搜索的路径相对RRT搜索的路径更优。

## 遇到的问题

本次作业的工作量不大，甚至需要修改的代码量加起来不超过50行，但是一份ROS工程比较大，里面有kdtree、visualization等模块。刚拿到作业的时候比较困惑，各个模块不会使用。本来打算阅读完所有模块的代码后再着手补全RRT\*的程序，后来慢慢觉得有些耗费时间，其实只需要在算法的最核心部分程序中理解各模块的用法，按照RRT\*的思路，利用各种工具函数补全程序就够了，补全RRT\*算法后发现比想象的简单很多。