机器人路径规划

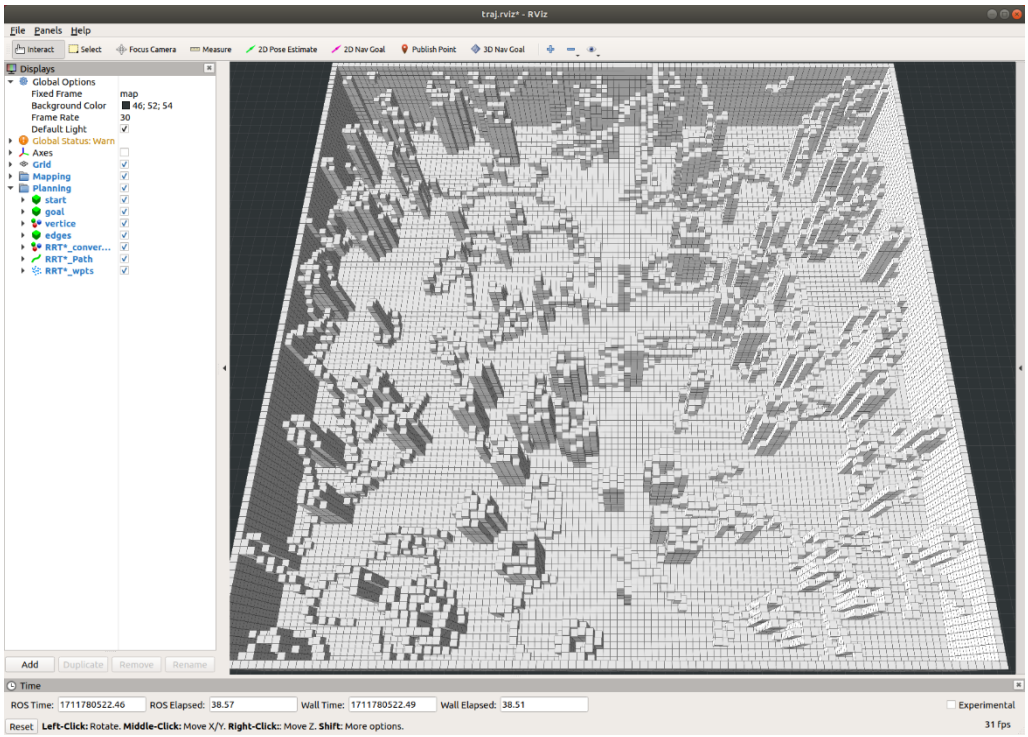
第三章作业

张 文泰

学号：21009101463

实验一　ROS 版本作业

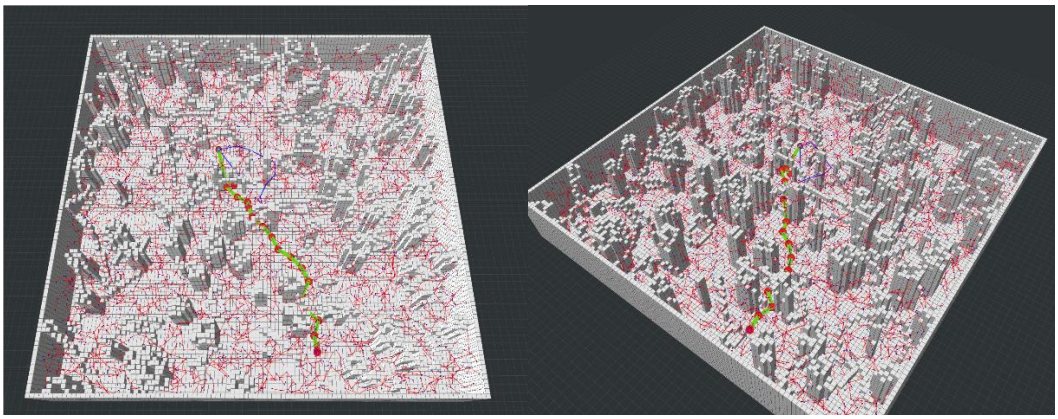
一、 实验过程



ROS 版本作业过程图

二、 实验结果

路径规划结果如图所示：



实验二　MATLAB 版本作业

一、代码补充与分析

```
for iter = 1:3000
    x_rand=Sample(Imp,goal);
    [x_near,index]= Near(x_rand,T);
    plot(x_near(1), x_near(2), 'ro', 'MarkerSize',2);
    x_new=Steer(x_rand,x_near,Delta);
    if ~collisionChecking(x_near,x_new,Imp)
        continue;
    end
    count=count+1;
    T.v(count).x = x_new(1);
    T.v(count).y = x_new(2);
    T.v(count).xPrev = x_near(1);
    T.v(count).yPrev = x_near(2);
    T.v(count).dist=Distance(x_new,x_near);
    T.v(count).indPrev = index;
    if Distance(x_new,goal) < Thr
        break;
    end
    line([x_near(1),x_new(1)],[x_near(2),x_new(2)]);
    pause(0.05);
end
```

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：

```
function X_rand=Sample(map,goal)
if unifrnd(0,1)<0.5
   X_rand(1)= unifrnd(0,1)* size(map,1);
   X_rand(2)= unifrnd(0,1)* size(map,2);
else
   X_rand=goal;
end
```

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。

```
function X_new=Steer(X_rand,X_near,StepSize)
theta = atan2(X_rand(1)-X_near(1),X_rand(2)-X_near(2));
X_new = X_near(1:2) + StepSize * [sin(theta)  cos(theta)];
```

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)。

```
function [X_near,index]=Near(X_rand,T)
min_distance=sqrt((X_rand(1)-T.v(1).x)^2+(X_rand(2)-T.v(1).y)^2);
for T_iter=1:size(T.v,2)
    temp_distance=sqrt((X_rand(1)-T.v(T_iter).x)^2+(X_rand(2)-T.v(T_iter).y)^2);
    if temp_distance<=min_distance
        min_distance=temp_distance;
        X_near(1)=T.v(T_iter).x;
        X_near(2)=T.v(T_iter).y;
        index=T_iter;
    end
end
```

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。

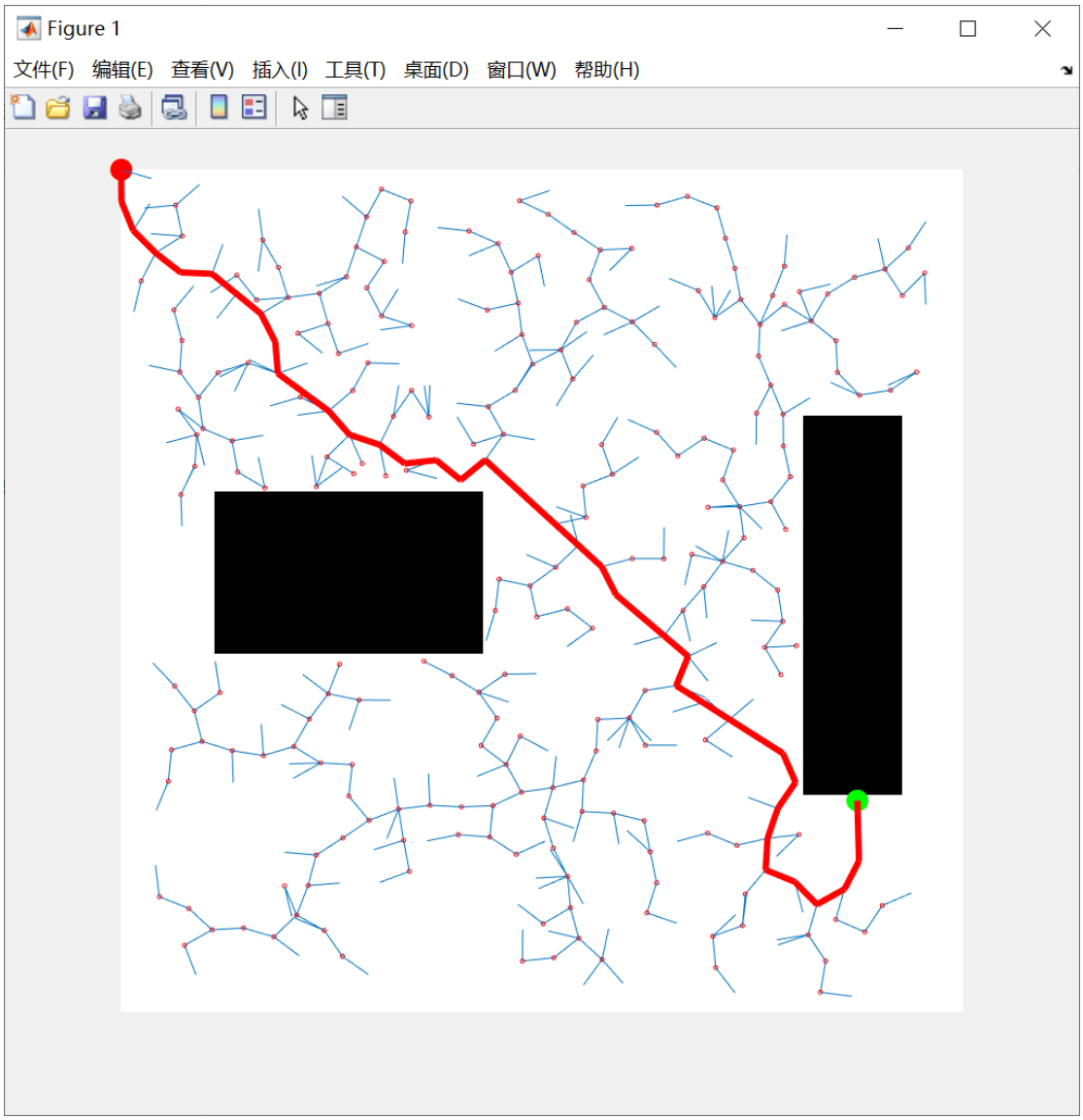
2. 遍历：遍历图中的所有点（通过 T.v 访问）。

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。

二、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。