



深蓝学院  
shenlanxueyuan.com

## 第三章作业分享



主讲人 世博



- 第一部分：数据结构
- 第二部分：RRT实现重点
- 第三部分：RRT\*实现重点
- 第四部分：优化尝试



# 第一部分 数据结构

## ●Matlab作业

当前节点横坐标	当前节点纵坐标	父节点横坐标	父节点纵坐标	当前节点到起始点的距离 (cost)	父节点索引
X	y	xPrev	yPrev	dist	indPrev

T.v

字段	x	y	xPrev	yPrev	dist	indPrev		
1	1	1	1	1	0	0		
2	19	24	1	1	29.2062	1		
3	42	42	1	1	57.9828	1		
4	46	71	21	32	83.2167	233		

与作业源码定义不符，我自己进行了修改



# 第二部分 RRT实现重点

## Rapidly-exploring Random Trees

### Algorithm 1: RRT Algorithm

**Input:**  $\mathcal{M}, x_{init}, x_{goal}$

**Result:** A path  $\Gamma$  from  $x_{init}$  to  $x_{goal}$

$\mathcal{T}.init();$

**for**  $i = 1$  **to**  $n$  **do**

$x_{rand} \leftarrow Sample(\mathcal{M});$

$x_{near} \leftarrow Near(x_{rand}, \mathcal{T});$

$x_{new} \leftarrow Steer(x_{rand}, x_{near}, StepSize);$

$E_i \leftarrow Edge(x_{new}, x_{near});$

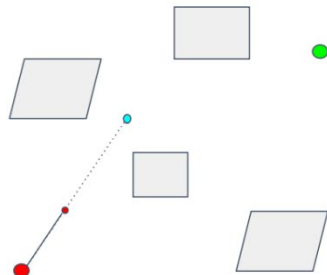
**if**  $CollisionFree(\mathcal{M}, E_i)$  **then**

$\mathcal{T}.addNode(x_{new});$

$\mathcal{T}.addEdge(E_i);$

**if**  $x_{new} = x_{goal}$  **then**

    Success();



**function**  $x_{new} = Steer(x_{rand}, x_{near}, StepSize)$

% 将距离随机点  $x_{rand}$  最近的节点  $x_{near}$  在  $x_{rand}$  方向上平移  $StepSize$  的距离, 生成新节点  $x_{new}$

$dis = distance(x_{near}, x_{rand});$

% 强迫症, 想让新节点坐标为整数, fix 舍余取整(也可不取整数)

$x_{new}(1) = fix(((dis - StepSize) * x_{near}(1) + StepSize * x_{rand}(1)) / dis);$

$x_{new}(2) = fix(((dis - StepSize) * x_{near}(2) + StepSize * x_{rand}(2)) / dis);$

**end**

%Step 1:在地图中随机采样一个点  $x_{rand}$   
%提示用  $(x_{rand}(1), x_{rand}(2))$  表示环境中采样点的坐标  
 $x_{rand} = randi(800, 1, 2);$  % 全局随机采样

% 用蓝色实心点表示采样点  
 $plot(x_{rand}(1), x_{rand}(2), 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'b');$

%Step 2: 遍历树, 从树中找到最近邻近点  $x_{near}$   
%提示:  $x_{near}$  已经在树  $T$  里  
 $[x_{near}, near\_idx] = Near(x_{rand}, T);$

%Step 3: 拓展得到  $x_{new}$  节点, 相似三角形原理  
 $x_{new} = Steer(x_{rand}, x_{near}, Delta);$

%Step 4: 检查节点是否是 collision free  
**if**  $\sim collisionChecking(x_{new}, x_{near}, Imp)$   
    **continue**;  
**end**

%Step 5: 将  $x_{new}$  插入  $T$ , 新节点  $x_{new}$  的父节点为  $x_{near}$   
 $T = AddNode(T, x_{new}, x_{near}, near\_idx);$

%Step 6: 检查是否达到目标点附近  
%提示:  $x_{new}$ ,  $x_{goal}$  之间的距离是否小于  $Thr$ , 小于则跳出 **for**  
 $dis\_goal = sqrt(power(x_{new}(1) - x_{goal}(1), 2) + power(x_{new}(2) - x_{goal}(2), 2));$

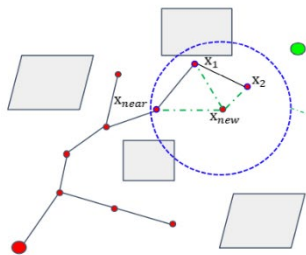
%Step 7: 将  $x_{near}$  和  $x_{new}$  之间的路径画出来  
 $X = [x_{near}(1), x_{new}(1)];$   
 $Y = [x_{near}(2), x_{new}(2)];$   
 $x\_steer = [x_{new}(1), x_{rand}(1)];$   
 $y\_steer = [x_{new}(2), x_{rand}(2)];$   
 $plot(X, Y, '-ob', x\_steer, y\_steer, 'r');$   
**hold on**;

%Step 8: 判断是否到达终点  
**if**  $dis\_goal < Thr$   
    **break**;  
**end**



# 第三部分 RRT\*实现重点

Rapidly-exploring Random Tree\*



Algorithm 2: RRT Algorithm

```
Input:  $\mathcal{M}, x_{init}, x_{goal}$   
Result: A path  $\Gamma$  from  $x_{init}$  to  $x_{goal}$   
 $T.init()$ ;  
for  $i = 1$  to  $n$  do  
   $x_{rand} \leftarrow Sample(\mathcal{M})$ ;  
   $x_{near} \leftarrow Near(x_{rand}, T)$ ;  
   $x_{new} \leftarrow Steer(x_{rand}, x_{near}, StepSize)$ ;  
  if CollisionFree( $x_{new}$ ) then  
     $X_{near} \leftarrow NearC(T, x_{new})$ ;  
     $x_{min} \leftarrow ChooseParent(X_{near}, x_{near}, x_{new})$ ;  
     $T.addNodeEdge(x_{min}, x_{new})$ ;  
     $T.rewire()$ ;
```

NearC: 以新节点 $x_{new}$ 为圆心，记录与新节点之间的cost小于radius的节点+最近节点 $x_{near}$ 的点集 $X_{near}$ 。

ChooseParent: 在节点集合 $X_{near}$ 中，以使得新节点到起点cost(dist)最小为基准，选择 $x_{new}$ 的父节点

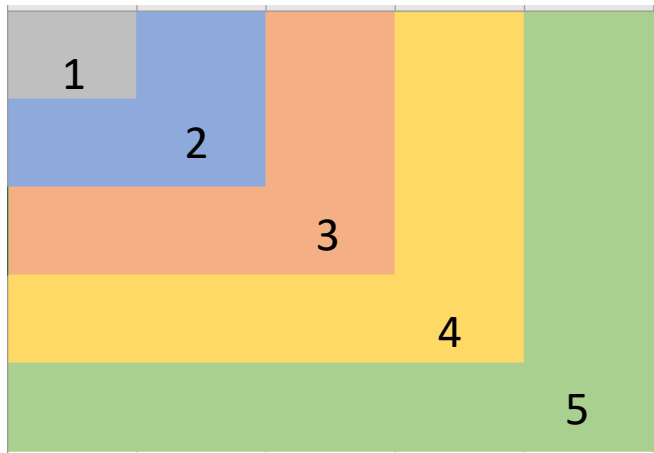
rewire: 遍历 $x_{new}$ 的邻近节点，查看它们通过 $x_{new}$ 节点到达起点的路径，是不是比它们之前的路径(dist)要短，如果是则将 $x_{new}$ 更新为它们的新父节点。



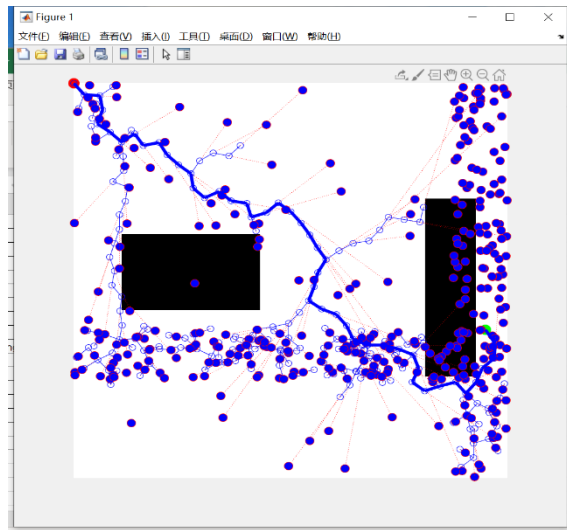
## 第四部分 优化尝试

RRT\*对于narrow-passage问题效果不佳，尝试修改采样函数进行优化

1. 首先对全部区域分区域，渐进全局采样



2. 对于终点所在区域重点采样

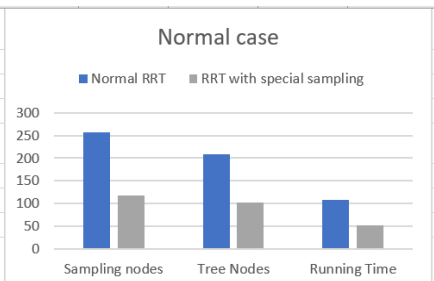




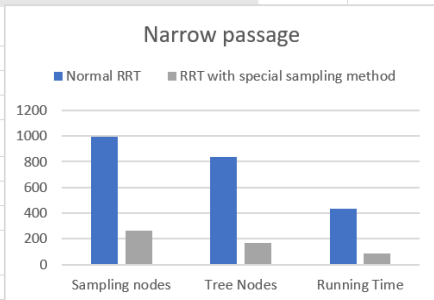
## 第四部分 优化尝试

### 效率对比

RRT	Sampling nodes	Tree Nodes	Running Time
Normal	332	276	142
	124	89	45
	316	262	134
Average	257	209	107
Narrow passage	1120	931	482
	1170	996	518
	686	594	308
Average	992	840	436



RRT with special sampling method	Sampling nodes	Tree Nodes	running time(s)
Normal	94	85	43
	165	136	70
	91	84	43
Average	117	102	52
Narrow passage	354	213	109
	184	152	78
	245	147	75
Average	261	171	87





# 在线问答

---

Q&A



感谢各位聆听 !  
Thanks for Listening

