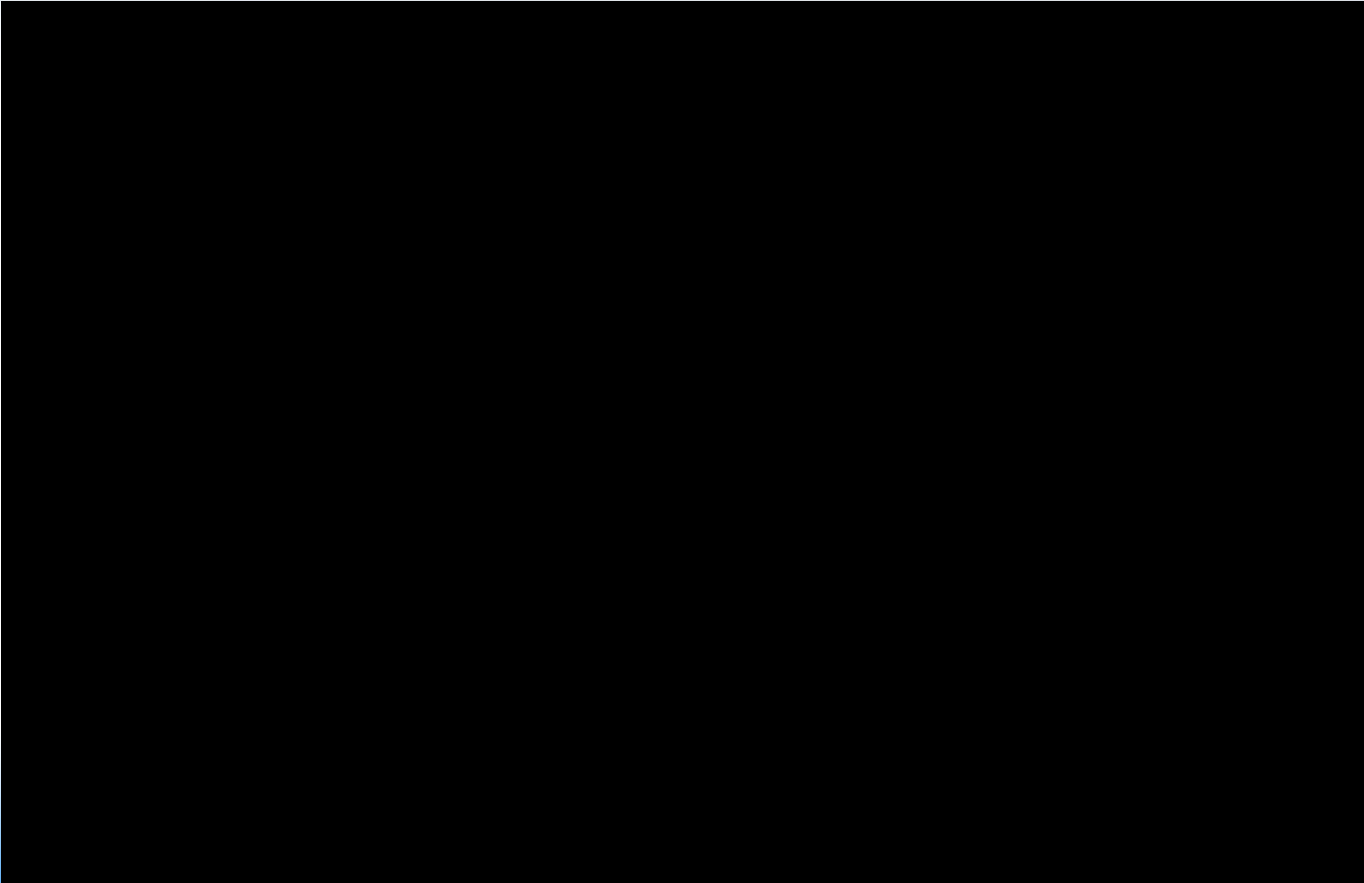




机器人路径规划

吴毅红，中国科学院大学，2020春季





2019年5月10号：
波士顿动力机
器人过独木桥

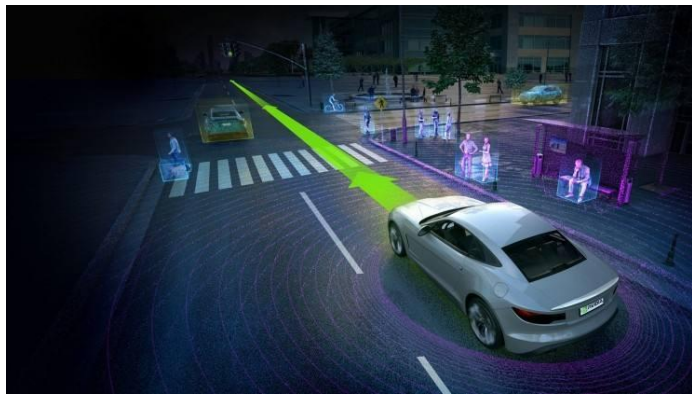


珞石机器人表演了一项自身引以为豪的技艺：在多个易拉罐之间仅1mm的空隙中飞速穿行环绕而不碰壁

路径规划

• 基本概念

路径规划技术，就是机器人根据自身传感器对环境的感知，**自行规划出一条安全的运行路线到达空间中的目标位置**。在智能移动机器人、自动驾驶、自动化机械臂控制等领域中起到了重要的作用。



路径规划分类

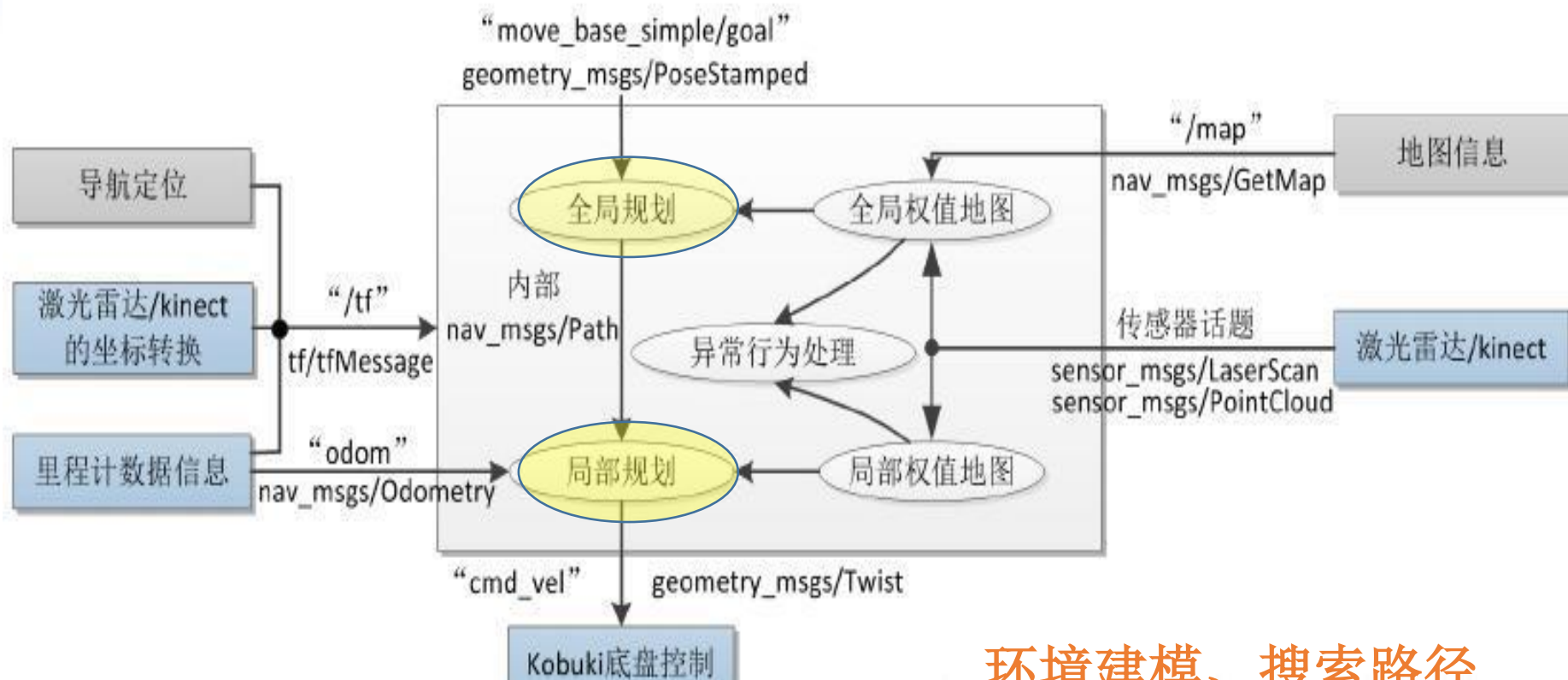
- 全局路径规划:

根据构建的环境地图，按照一定的算法搜寻一条最优或者次优的无碰撞路径，规划的精确程度和获取环境信息的准确程度有关

- 局部路径规划:

环境信息未知或者部分未知，根据传感器的信息来连续地更新局部的环境信息，从而确定出机器人在地图中的当前位置周围的障碍物分布情况，并在此基础上规划出一条从当前点到某一子目标的最优路径。

- 混合路径规划



环境建模、搜索路径

• 常见的全局路径规划算法

PRM (Probabilistic roadmaps)

RRT (Rapidly-exploring random tree)

D* (Dynamic A*)

算法类型	示例	是否完备	是否最优
基于搜索	A*, D*, Dijkstra	完备	最优(计算量大)
基于采样	PRM, RRT	概率完备	非最优 (如RRT) / 渐进最优 (如PRM*, RRT*, 快速)
基于启发	遗传, 蚁群	完备	非最优

完备性 (completeness)：是指如果在起始点和目标点间有路径解存在，那么一定可以得到解，如果得不到解那么一定说明没有解存在；

概率完备性 (probabilistic completeness)：是指如果在起始点和目标点间有路径解存在，只要规划或搜索的时间足够长，就一定能确保找到一条路径解；

最优性 (optimality)：是指规划得到的路径在某个评价指标上是最优的（评价指标一般为路径的长度）；

渐进最优性 (asymptotic optimality)：是指经过有限次规划迭代后得到的路径是接近最优的次优路径，且每次迭代后都与最优路径更加接近，是一个逐渐收敛的过程；

路径规划发展趋势

- 多传感器融合
- 多机器人系统
- 复杂环境高维自由度
- 人机共融

2018年3月15日，腾讯宣布其将成立机器人实验室
“Robotics X”，并引进张正友
2017年，小布机器人；小Q机器人

- Steven M. LaValle, James J. Kuffner, Jr. Randomized Kinodynamic Planning. The International Journal of Robotics Research, 20(5): 378-400, 2001. (RRT)
- Sertac Karaman, Emilio Frazzoli. Incremental Sampling-based algorithms for optimal motion planning.
<http://arxiv.org/abs/1005.0416>. (RRT*)

Randomized Kinodynamic Planning(RRT)

基本关键概念

- Kinodynamic Planning
- Configuration Space
- State Space
- Metric
- Holonomic
- Nonholonomic

这就是一种非完整性等式约束，意味着机器人的瞬时运动方向是两个自由度方向而不是三个，且限制了机器人的行驶方向即任意时刻机器人的运动方向只能与机器人的当前朝向相一致。文献[7]中证明了上式是不可积分的所以该式是非完整等式约束。一种非完整性等式约束限制了机器人的速度空间是在 m 维构型空间的 $m-1$ 子空间。

完整性约束只限制受控系统的空间位置，或者同时限制空间位置和运动速度但可以通过积分全部转化为对空间位置的约束，其物理意义在于系统不能在任意时刻取空间中任意位置。假设机器人 R 在 m 维构型空间 C 中运动， q 为机器人在 m 维空间中的位姿，则完整性约束一般可表示为：

$$F(q, t) = F(q_1, \dots, q_m, t) = 0 \quad (1.3)$$

其中 F 是非零导数的光滑函数。上面的约束条件决定了 m 维空间中机器人的可行位置空间。上面公式表示的完整性约束叫做完整性等式约束条件。

非完整性约束条件是同时限制受控系统的空间位置和运动速度，并且速度项不能通过积分变换转换为空间位置的约束条件，也称之为不可积约束。具有非完整约束的系统就称之为非完整控制系统，由于其约束的不可积性，使其控制与规划变得相当的困难与复杂。

非完整性等式约束可表示为：

$$G(q, \dot{q}, t) = G(q_1, \dots, q_m, \dot{q}_1, \dots, \dot{q}_m, t) = 0 \quad (1.5)$$

其中 G 是光滑函数， $q_i = \frac{dq_i}{dt} (i=1, \dots, m)$ ，则 $\dot{q} = (\dot{q}_1, \dots, \dot{q}_m)$ 是机器人在 q 的速度向量。公式中的速度项是不可积分的，若全部可积则就可以变成完整性约束，如果部分可积分可

利用矩阵变化消去可积部分。

Incremental Sampling-based algorithms for optimal motion planning (RRT*)

基本关键概念:

- RRG
- Completeness
- Probabilistic Completeness
- Optimality
- Asymptotic Optimality

• RRT算法

Algorithm 1: Body of RRT and RRG Algorithms

```

1  $V \leftarrow \{x_{init}\}; E \leftarrow \emptyset; i \leftarrow 0;$ 
2 while  $i < N$  do
3    $G \leftarrow (V, E);$ 
4    $x_{rand} \leftarrow \text{Sample}(i); i \leftarrow i + 1;$ 
5    $(V, E) \leftarrow \text{Extend}(G, x_{rand});$ 

```

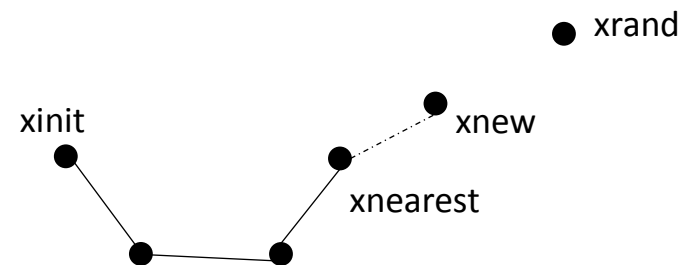
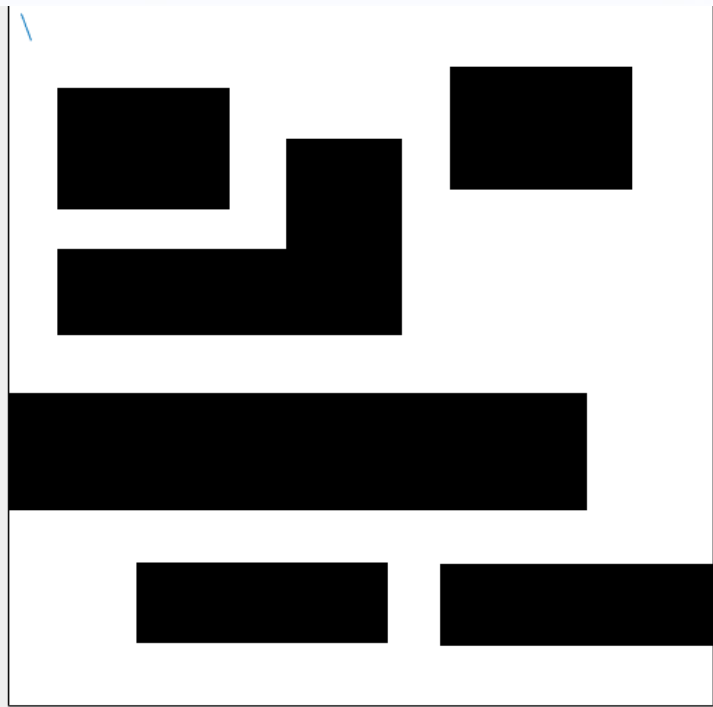
Algorithm 2: Extend_{RRT}

```

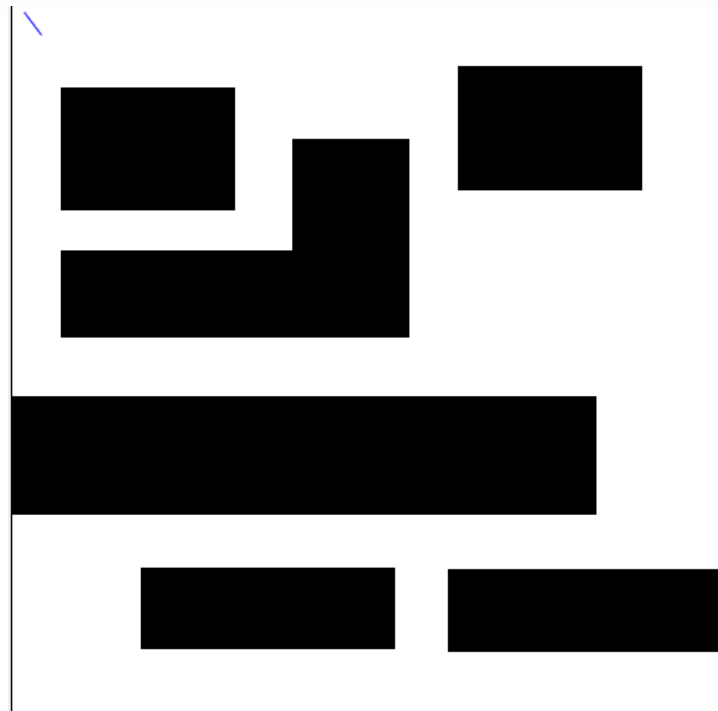
1  $V' \leftarrow V; E' \leftarrow E;$ 
2  $x_{nearest} \leftarrow \text{Nearest}(G, x);$ 
3  $x_{new} \leftarrow \text{Steer}(x_{nearest}, x);$ 
4 if  $\text{ObstacleFree}(x_{nearest}, x_{new})$  then
5    $V' \leftarrow V' \cup \{x_{new}\};$ 
6    $E' \leftarrow E' \cup \{(x_{nearest}, x_{new})\};$ 
7 return  $G' = (V', E')$ 

```

1. **初始化**时随机树 T 只包含一个节点：根节点 x_{init} 。
 2. 首先**Sample**函数从状态空间中随机选择一个**采样**点 x_{rand}
 3. 然后**Nearest**函数从随机树中选择一个距离 x_{rand} **最近**的节点 $x_{nearest}$;
 4. 最后**Steer**函数通过从 $x_{nearest}$ 向 x_{rand} **扩展**一段距离，得到一个新的节点 x_{new} 。
 5. 如果 x_{new} 与障碍物**没有发生碰撞**，则将 q_{new} **加入**到随机树中。
- 重复上述步骤直到树中的节点包含有到目标点 q_{goal} 距离小于一个阈值的点。



- RRT算法的简单升级
- 1. 建立两个树，分别以 x_{init} 和 x_{goal} 为初始化节点。轮流把对方的 x_{new} 放入自己 $nearest$ 函数中。这种双向的RRT技术具有良好的搜索特性，比原始RRT算法的搜索速度、搜索效率有了显著提高，被广泛应用。
- 2. 在随机树每次的生长过程中，根据随机概率来决定 q_{rand} 是目标点还是随机点。在 $Sample$ 函数中设定参数 $Prob$ ，每次得到一个0到1.0的随机值 p ，当 $0 < p < Prob$ 的时候，随机树朝目标点生长行；当 $Prob < p < 1.0$ 时，随机树朝一个随机方向生长。



• RRT*

Algorithm 4: Extend_{RRT*}

```

1  $V' \leftarrow V; E' \leftarrow E;$ 
2  $x_{\text{nearest}} \leftarrow \text{Nearest}(G, x);$ 
3  $x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}}, x);$ 
4 if  $\text{ObstacleFree}(x_{\text{nearest}}, x_{\text{new}})$  then
5    $V' \leftarrow V' \cup \{x_{\text{new}}\};$ 
6    $x_{\text{min}} \leftarrow x_{\text{nearest}};$ 
7    $X_{\text{near}} \leftarrow \text{Near}(G, x_{\text{new}}, |V|);$ 
8   for all  $x_{\text{near}} \in X_{\text{near}}$  do
9     if  $\text{ObstacleFree}(x_{\text{near}}, x_{\text{new}})$  then
10        $c' \leftarrow \text{Cost}(x_{\text{near}}) + c(\text{Line}(x_{\text{near}}, x_{\text{new}}));$ 
11       if  $c' < \text{Cost}(x_{\text{new}})$  then
12          $x_{\text{min}} \leftarrow x_{\text{near}};$ 
13    $E' \leftarrow E' \cup \{(x_{\text{min}}, x_{\text{new}})\};$ 
14   for all  $x_{\text{near}} \in X_{\text{near}} \setminus \{x_{\text{min}}\}$  do
15     if  $\text{ObstacleFree}(x_{\text{new}}, x_{\text{near}})$  and
16        $\text{Cost}(x_{\text{near}}) >$ 
17        $\text{Cost}(x_{\text{new}}) + c(\text{Line}(x_{\text{new}}, x_{\text{near}}))$  then
18        $x_{\text{parent}} \leftarrow \text{Parent}(x_{\text{near}});$ 
19        $E' \leftarrow E' \setminus \{(x_{\text{parent}}, x_{\text{near}})\};$ 
20        $E' \leftarrow E' \cup \{(x_{\text{new}}, x_{\text{near}})\};$ 
21 return  $G' = (V', E')$ 

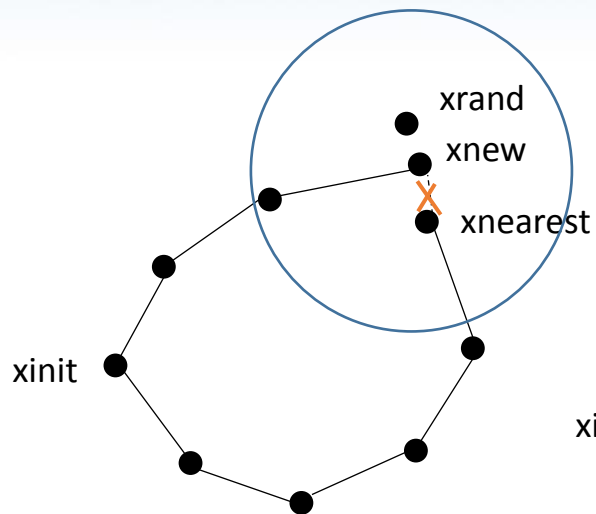
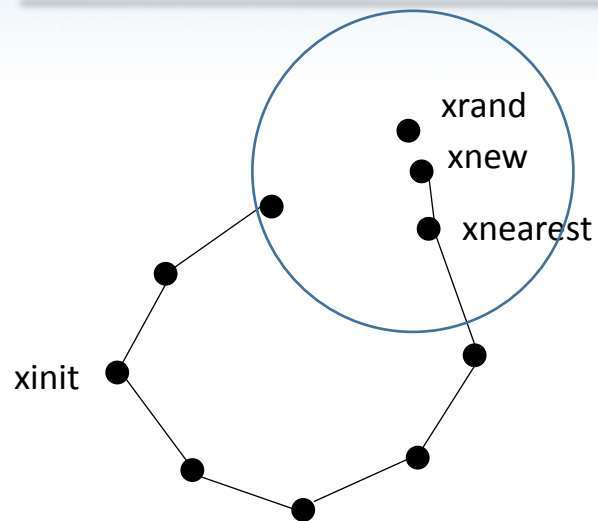
```

在RRT的基础上增加了对树的“修剪整理”操作。在把 x_{new} 加入树前检查以哪个节点为父节点得到的总cost最小。在 x_{new} 加入后检查是否别的节点以 x_{new} 为父节点得到的总cost更小。

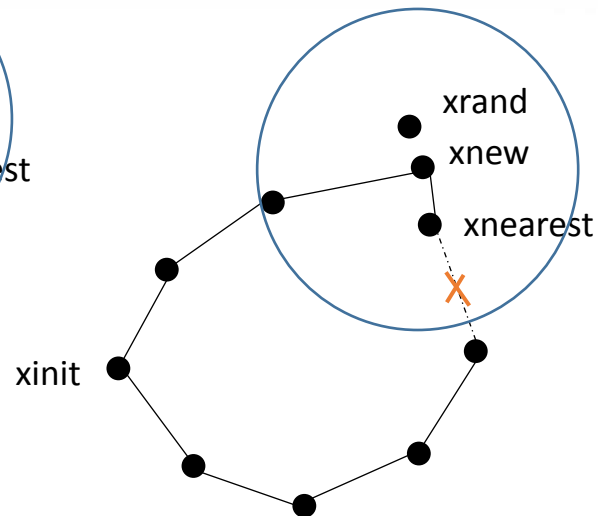
q_{new} 为中心生成一个半径为 R 的圆，在圆里找到点半径小于 R 的点（每个点有相应的cost，cost代表他从根节点到当前点的代价），

如果 $\text{dist}(\text{point1}, q_{\text{new}}) + \text{cost}(\text{point1}) < \text{cost}(\text{point2}) + \text{dist}(\text{point1}, q_{\text{new}})$ ，则 q_{new} 的父节点就为 point1 ，

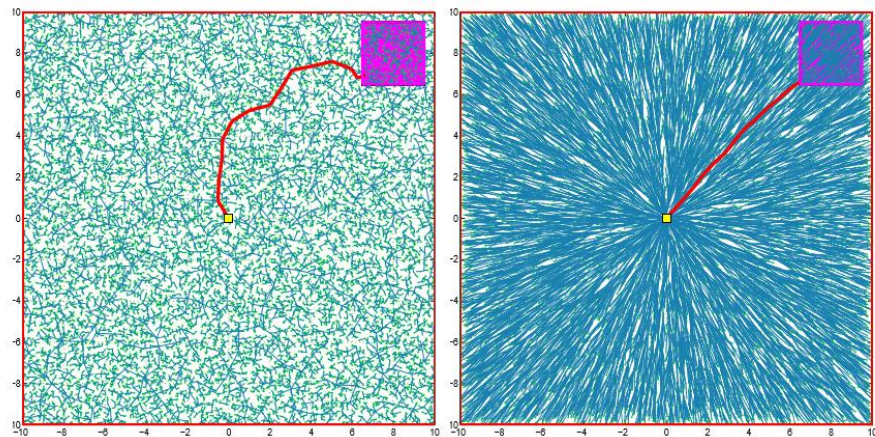
如果 $\text{dist}(\text{point2}, q_{\text{new}}) + \text{cost}(q_{\text{new}}) < \text{cost}(\text{point2})$ ，则更改 point2 的父节点为 q_{new} 。



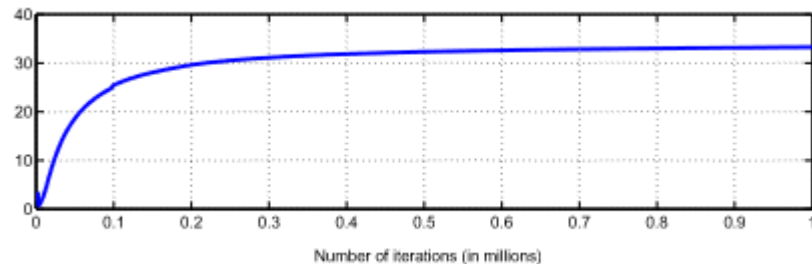
Xnew以别的点为父节
点cost会更低



别的点以Xnew为父节
点cost会更低



因此RRT*比RRT产生的轨迹质量更高。



但是相应的算法复杂度也更高（30多倍）。

	是否完备	是否最优
RRT	概率完备	非最优
RRT*	概率完备	渐进最优

课后作业

了解BAT在机器人领域的布局，以及它们各自的特色

消费级机器人公司优必选有哪些消费级机器人？

你对人形机器人的前景如何看待？

谢 谢

yhwu@nlpr.ia.ac.cn
<http://vision.ia.ac.cn/>

