

# 路径规划常用算法

## 一、几何方法

可视图、切线图、Voronoi图、精确栅格分解

- 1. 可视图： 在空间中，以多边形障碍物模型为基础，任意形状障碍物用近似多边形代替，用直线将机器人运动的起始点和空间内障碍物的顶点以及目标点连接，并保证这些直线段不与空间障碍物相交，形成一张图，即可视图。然后采用图搜索算法寻找从起始点到目标点的最优路径，搜索最优路径就转化为起始点到目标点经过这些可视直

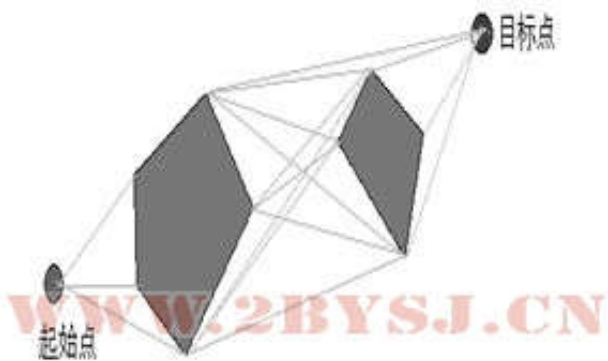


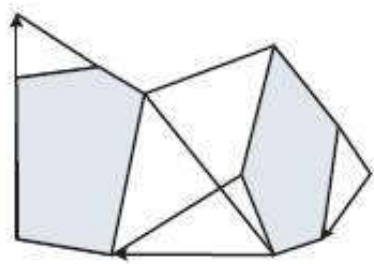
图 2-1 可视图法示意图

线的最短距离问题。

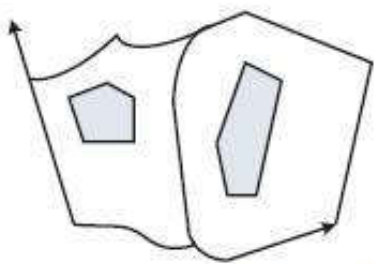
畢業設計 WWW.2BYSJ.CN

- 优点：直观，容易求得最短路径。
- 缺点：缺乏灵活性，局部路径规划能力差；存在组合爆炸问题。

- 2. 切线图：以多边形障碍物模型为基础, 任意形状障碍物用近似多边形替代, 在自由空间中构造切线图。因此从起始点到目标点机器人是沿着切线行走，即机器人必须几乎接近障碍物行走，路径较短。



(a) 切线图

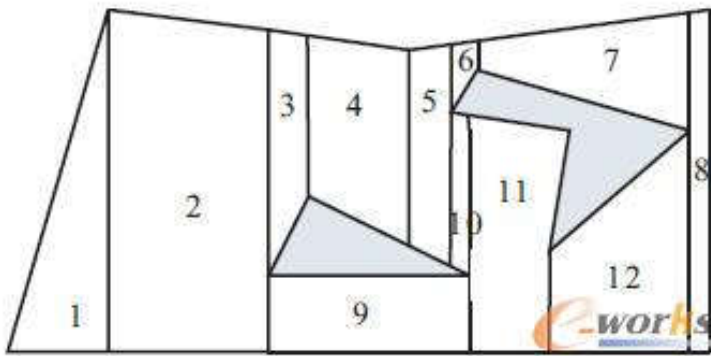


(b) Voronoi图orks

- 3. Voronoi图：Voronoi图由一系列的直线段和抛物线段构成，直线由两个障碍物的顶点或两个障碍物的边定义生成，直线段上所有点必须距离障碍物的顶点或障碍物的边相

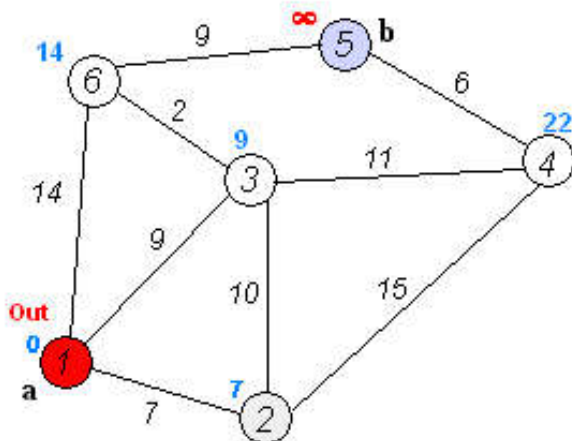
等，抛物线段由一个障碍物的顶点和一个障碍物的边定义生成，抛物线段同样要求与障碍物顶点和障碍物的边有相同距离，与切线法相比，Voronoi图法从起始节点到目标节点的路径将会增长，但采用这种控制方式时，即使产生位置误差，移动机器人也不会碰到障碍物，安全性较高。

4. 栅格法：将机器人周围空间分解为相互连接且不重叠的空间单元（栅格），由这些栅格构成一个连通图，依据障碍物占有情况，在此图上搜索一条从起始栅格到目标栅格无碰撞的最佳路径。



## 二、前向图搜索算法

1. **Dijkstra**算法：在显式图中找到从源节点到任意节点的最短路径。如果该算法在到达目标节点时结束，那么一定可以找一条从源节点到目标节点的最短路径。  
设 $G=(V,E)$ 是一个带权有向图，把图中顶点集合 $V$ 分成两组，第一组为已求出最短路径的顶点集合（用 $S$ 表示，初始时 $S$ 中只有一个源点，以后每求得一条最短路径，就将加入到集合 $S$ 中，直到全部顶点都加入到 $S$ 中，算法就结束了），第二组为其余未确定最短路径的顶点集合（用 $U$ 表示），按最短路径长度的递增次序依次把第二组的顶点加入 $S$ 中。在加入的过程中，总保持从源点 $v$ 到 $S$ 中各顶点的最短路径长度不大于从源点 $v$ 到 $U$ 中任何顶点的最短路径长度。此外，每个顶点对应一个距离， $S$ 中的顶点的距离就是从 $v$ 到此顶点的最短路径长度， $U$ 中的顶点的距离，是从 $v$ 到此顶点只包括 $S$ 中的顶点为中间顶点的当前最短路径长度。



优点：成功率高，鲁棒性好；

缺点：效率低，遍历节点多。

2. **A\***算法: 根据估计函数 $f=g+h$  ( $g$ 表示起点到指定点的路程,  $h$ 表示从指定点到终点的路程) 依次寻找 $f$ 值最小的点。

优点：扩展节点少，鲁棒性好；

缺点：搜索的复杂度高，计算量大。

3. **D\***算法: ①先用Dijkstra算法从目标节点 $G$ 向起始节点搜索。储存路网中目标点到各个节点的最短路程 $h$ 。

②机器人沿最短路径开始移动, 当在 $Y$ 点探测到下一节点 $X$ 状态发生改变, 如堵塞。机器人调整自己在当前位置 $Y$ 到目标点 $G$ 的实际值 $h(Y)$ ,  $h(Y)=X$ 到 $Y$ 的新权值 $c(X,Y)+X$ 的原实际值 $h(X)$ , 再重新找 $Y$ 点附近 $h$ 值最小的点。

4. 人工势场法: 将无人机在周围环境中的运动, 假想成引力场中的运动, 目标点对无人机产生“引力”, 障碍物对无人机产生“斥力”, 最后通过求合理来控制无人机的运动。

优点：规划出的路径平滑且安全；

缺点：容易陷入局部极小点，到不了目标位置；不适合寻找最短路径。

### 三、基于随机采样的运动规划方法

1. **RPP** (随机路径规划器): 在人工势场法的基础上通过执行随机步骤使其摆脱局部极小。

定义  $h(v)$ 代表吸引势和排斥势合成的启发式势函数 (从某节点至目标点的费用估计)。

对于RPP规划器来说,定义三种模式: DESCEND,ESCAPE和BACKTRACK。最初时,规划器处于DESCEND模式, 从 $V_{init}$ 开始按照梯度下降的原则产生新节点。在DESCEND模式下,在图 $G$ 中产生一个新节点 $V_e$ , 在 $V_e$ 的邻域内且使得 $h(v)$ 最小的方向上产生一个新节点 $V_n$ 作为规划的下一个节点。如果陷入局部极小,通过梯度下降产生新节点的操作会失败。此时, 规划器转入 ESCAPE模式,试图摆脱局部极小。在 ESCAPE模式下, 执行一个随机步骤创建新节点, 结束后转向 DESCEND 模式。如果在ESCAPE 模式下, 一定步数内,连续随机产生新节点失败,规划器将转入BACKTRACK模式。在此模式下, 在图 $G$ 中,从上次产生的最新节点到图 $G$ 的起始节点之间随机选择一个节点,从该节点开始规划器进入ESCAPE模式。

优点：克服了局部极小的限制；

缺点：参数难调。

2. **PRM** (概率路标算法): 其主要思想是用一个随机路标图来表示机器人系统运行的自由空间 $C_{free}$ ,然后在此图中为机器人系统搜索到一条可行的路径。

一般的 PRM 规划方法的操作过程大体可分为学习阶段(预处理阶段)和搜寻阶段。学习阶段的主要目的是在C free中为机器人建立一张地图。在该阶段,首先以随机的方式在C free中选取大量的机器人位姿,构成地图中的节点集 $N$ 。然后由一个局部规划器来为 $N$ 中的每个节点寻找它们的近邻节点,并为它们建立连接,从而构成地图中的边集 $E$ 。经过学习阶段建立起来的地图是一张较大的无向图 $R=(N,E)$ 。

在搜索阶段, 首先将规划任务的起始点 $q_{init}$ 和目标点 $q_{goal}$ 加入节点集 $N$ ,并由局部规划器为其寻找近邻节点并建立连接, 从而在地图 $R$ 中形成了若干个连通图。如果起始点和目标点在同一个连通图中, 则存在机器人运动的可行路径,然后在地图  $R$  中采用某种搜索算法在起始点 $q_{init}$ 和目标点 $q_{goal}$ 之间搜索一条通路, 即为机器人的可行路径。如果起始点和目标点不处于同一个连通图中, 则回到学习阶段进行节点扩充操作, 来增加地图在C free的“困难地区(如狭窄通道等)”的连通性。

3. **RRT(快速随机搜索树算法)**: 基本思想是以初始点作为根节点, 通过随机采样增加叶子节点的方式, 生成一个随机扩展树, 当随机树种的叶子节点包含了目标点时, 随机树中找到一条由初始点到目标点的路径。

## 四、其他智能化规划方法

1. 遗传算法: 首先初始化路径群体, 然后进行遗传操作, 如选择、交叉、复制、变异。经过若干代的进化以后, 停止进化, 输出当前最优个体作为路径下一个节点。

优点: 可以获得较短的路径。

缺点: 进化速度难以控制, 参数难调。

2. 模糊规划器: 模糊规划器利用反射式导航机制, 将当前环境障碍信息作为模糊推理器的输入, 推理器输出机器人的转向角和速度等。

优点: 在环境未知或发生变化的情况下, 能够快速而准确地规划机器人局部路径, 对于要求有较少路径规划时间的机器人是一种很好的导航方法。

缺点: 当障碍物数目增加时, 计算量较大; 只利用局部信息作为快速反应, 较容易陷入局部极小。

3. 神经网络: 基本原理是将环境障碍等作为神经网络的输入层信息, 经由神经网络并行处理, 神经网络输出层输出期望的转向角和速度等, 引导机器人避障形式, 直至到达目的地。

优点: 并行处理效率高; 具有学习功能; 能收敛到最优路径; 易于从二维空间推广到三维空间。

缺点: 随着机器人自由度的增加和环境复杂度增强, 存在效率问题; 难以获得最短路径。

4. 强化学习（**Q-Learning**）：一种无监督学习算法，让一个Agent从起点开始寻找路径，路径正确时给予奖励，错误是给予惩罚，更新评估函数，重复循环迭代学习过程，直到满足整个学习的条件。