智能汽车路径规划与轨迹跟踪系列算法精讲及Matlab程序实现

第4讲　A*算法

创作者：Ally

时间：2021/1/11

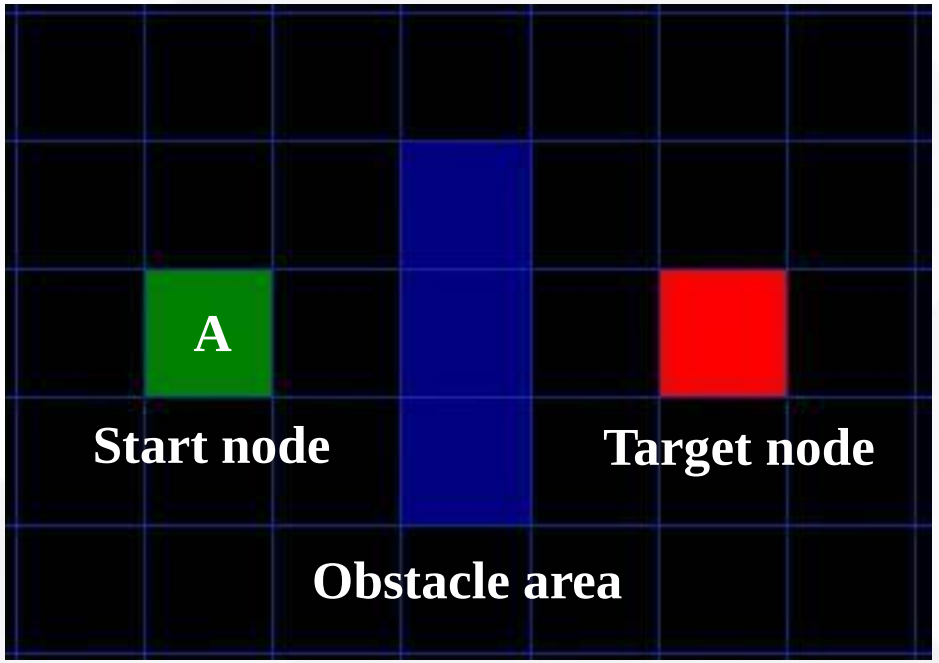
1 学习课程大纲目录


路径规划与轨迹跟踪系列学习视频
自动驾驶汽车定位-感知-规划-决策-控制概述
全局路径规划
Dijkstra算法
蚁群算法
动态规划算法
A*算法
局部路径规划
多项式曲线法
势场法
贝塞尔曲线
B样条曲线
轨迹跟踪与控制
纯跟踪法
Stanley法
PID法
MPC法

2 全局路径规划算法——A*算法

算法简介

- A*（A-Star)算法是一种静态路网中求解最短路径最有效的直接搜索方法，也是解决许多搜索问题的有效算法。
- 广泛应用于室内机器人路径搜索、游戏动画路径搜索等



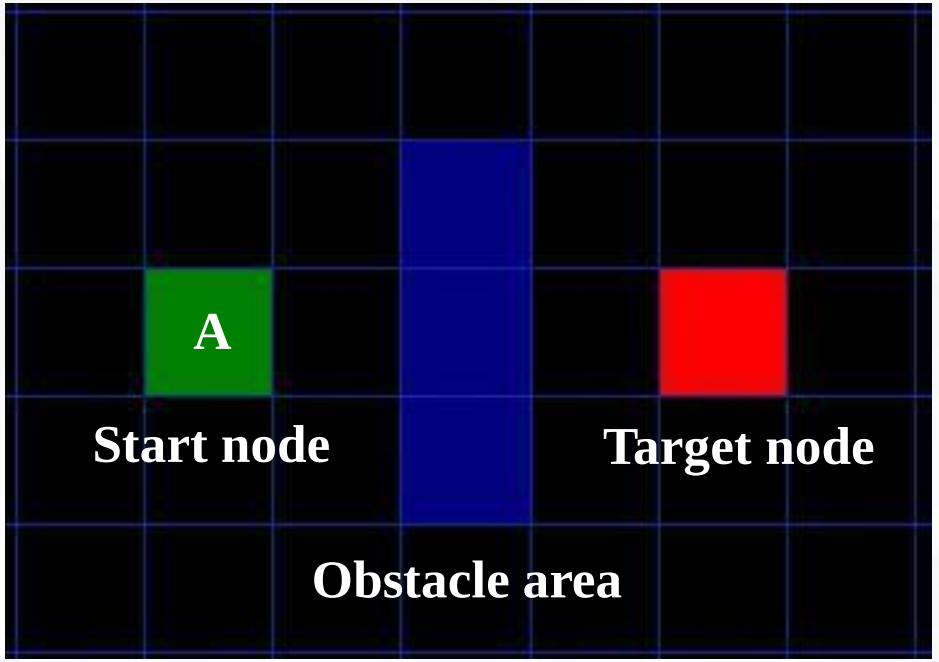
算法思想

- A*算法结合了贪心算法（深度优先）和Dijkstra算法（广度优先），是一种启发式搜索算法。
- 路径优劣评价公式为： f(n)=g(n)+h(n)。
- f(n) 是从初始状态经由状态n到目标状态的代价估计，
  g(n) 是在状态空间中从初始状态到状态n的实际代价，
  h(n) 是从状态n到目标状态的最佳路径的估计代价。
- 使用了两个状态表，分别称为openList表和closeList表。openList表由待考察的节点组成， closeList表由已经考察过的节点组成。

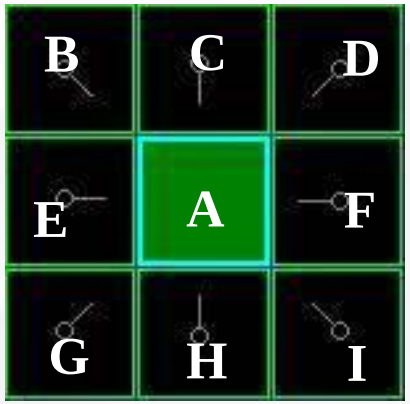
2 全局路径规划算法——A*算法

算法精讲—预处理

- 将地图栅格化，把每一个正方形格子的中央称为节点；
- 确定栅格属性，即每一个格子有两种状态：可走和不可走（体现障碍物）。
- 定义两个列表集合：openList和closeList。openList表由待考察的节点组成， closeList表由已经考察过的节点组成。类似Dijkstra算法的U集合和S集合。
- 确定起始节点和目标节点。

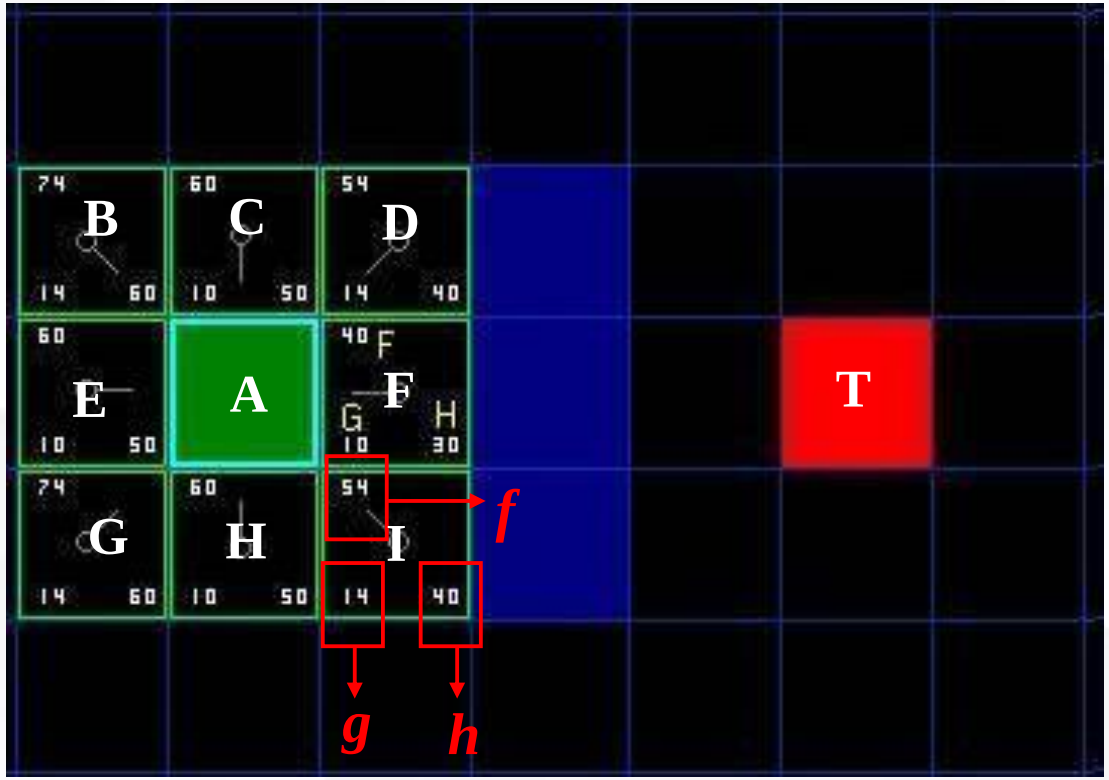


- 初始时，定义A为父节点，节点A离自身距离为0，路径完全确定，移入closeList中；
- 父节点A周围共有8个节点，定义为子节点。将子节点放入openList中，成为待考察对象。
- 若某个节点既未在openList，也没在closeList中，则表明还未搜索到该节点。
- 路径优劣判断依据是移动代价，单步移动代价采取Manhattan 计算方式，即把横向和纵向移动一个节点的代价定义为10。斜向移动代价参考等腰三角形计算斜边的方式，距离为14。



算法精讲—开始搜索

- 移动代价评价函数为： f(n)=g(n)+h(n)。f(n) 是从初始状态经由状态n到目标状态的代价估计，g(n) 是在状态空间中从初始状态到状态n的实际代价，h(n) 是从状态n到目标状态的最佳路径的估计代价。以节点I为例。
- 首先考察g，由于从A到该格子是斜向移动，单步移动距离为14，故g = 14.
- 再考察估计代价h。估计的含义是指忽略剩下的路径是否包含有障碍物（不可走），完全按照Manhattan计算方式，计算只做横向或纵向移动的累积代价：横向向右移动3步，纵向向上移动1步，总共4步，故为h = 40.
- 因此从A节点移动至I节点的总移动代价为： f=54
- 以此类推，分别计算当前openList中余下的7个子节点的移动代价，挑选最小代价节点F，移到closeList中。
- 现在openList = {B,C,D,E,G,H,I}, closeList = {A,F}



算法精讲—继续搜索

- 从openList中选择 f值最小的（方格）节点I，从 openList 里取出，放到 closeList 中，并把它作为新的父节点 。
- 检查所有与它相邻的子节点，忽略障碍物不可走节点、忽略已经存在于closeList的节点；如果方格不在openList中，则把它们加入到 openList中。
- 如果某个相邻的节点已经在 open list 中，则检查这条路径是否更优，也就是说经由当前节点(我们选中的节点) 到达那个节点是否具有更小的 G 值。如果没有，不做任何操作。
- 依次类推，不断重复。一旦搜索到目标节点T，完成路径搜索，结束算法。