

# Motion Planning for Mobile Robots - HW2 ROS 版

peng00bo00

June 20, 2020

## 1. A\* 算法流程如下:

- (a) 初始化起始点并将起始点加入到 openSet 中;
- (b) While openSet 非空:
  - 从 openSet 中选择 fScore 最小的点将其删除并加入到 closeSet 中;
  - 若该结点为终止结点, 算法结束;
  - 遍历当前结点所有的相邻结点
    - i. 若相邻节点既不在 openSet 也不在 closeSet 中, 则初始化相邻结点并加入到 openSet 中;
    - ii. 若相邻节点在 openSet 中, 若相邻结点的 fScore 可以减小则对其进行更新, 否则跳过该结点继续循环;
    - iii. 若相邻节点在 closeSet 中, 则跳过该节点继续循环;

## 2. 实现的 A\* 算法如 Fig.1所示

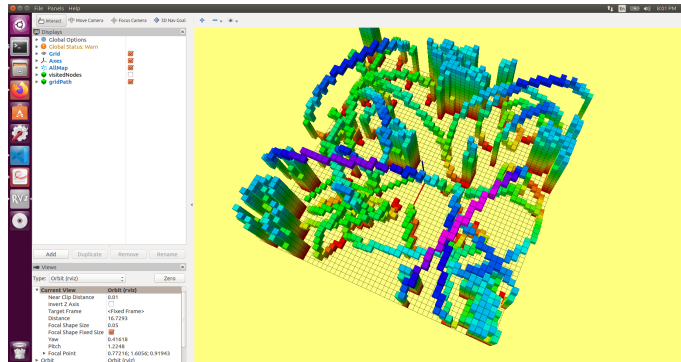


Figure 1: A\* 算法寻找最优路径

测试从地图 4 个角点出发的计算耗时, 使用不同 Heuristic Function 的效率对比如 Table.1所示。结果显示采用不同 Heuristic Function 的运行耗时均在毫秒级, 其中 Manhattan Distance 的用时最短而 Diagonal Distance 的用时最长。

Table 1: A\* 算法运行效率 (ms)

Heuristic Function	$t_1$	$t_2$	$t_3$	$t_4$
Euclidean	1.417	1.976	2.235	1.789
Manhattan	0.088	0.098	0.159	0.153
Diagonal	6.998	8.287	6.714	8.096

加入了 Tie Breaker 后的效率对比如 Table.2所示。结果显示加入 Tie Breaker 后可能会略微改善计算效率, 而在大多数情况下对计算效率影响不大, 这可能与地图的具体形式有关。

Table 2: A\* 算法 +Tie Breaker 运行效率 (ms)

	$t_1$	$t_2$	$t_3$	$t_4$
Euclidean	3.632	3.014	1.274	0.702
Manhattan	0.179	0.072	0.102	0.068
Diagonal	6.450	6.633	6.176	8.721

JPS 相关代码可参见 src/grid\_path\_searcher/src/readonly/JPS\_searcher.cpp