

(一) 算法流程: 主要说明 AstarGraphSearch 函数中的实现

(1) 实现 step 1 中的要求, 即完成 getHeu 函数, 在这个函数里接受两个指针参数 GridNodePtr, 分别对他们的坐标求欧式距离或者曼哈顿距离;

(2) 实现 step 2 中的要求, 把在初始化函数中已经初始化好的 GridNodeMap 中每一个 node 的 fscore 设置好; 并把 GridNodeMap 中代表路径起点的 node 的 id 设置为 1;

(3) 实现 step 3 中的要求, 将 currentPtr 设置为 openset 中的第一个, 并把第一个从 openset 中删除;

(4) 实现 step 4 中的要求, 即实现 AstarGetSucc 函数, 将 currentPtr 的 index 提取出来, 找到 GridNodeMap 中此 index 在 xyz 轴前后移动一个单位的 node。用 isfree 判断是否非障碍和在地图范围内, 判断为 true 则更新 node 的属性后加入到 neighborPtrSets 中。在我的实现中没用到 edgeCostSets。

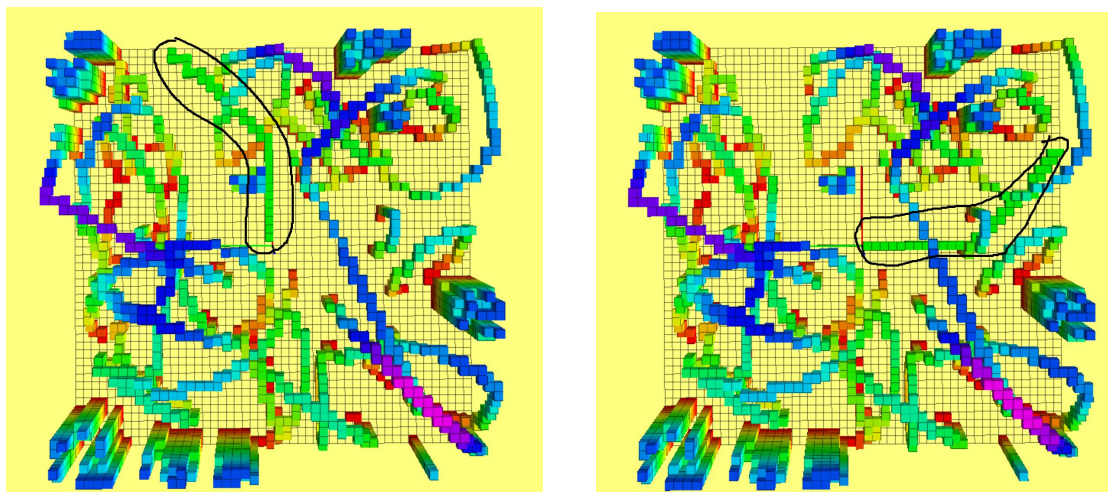
(5) 实现 step 5 和 6 中的要求, 当 id=0 时创建一个新的 GridNodePtr 并加入到 openset 中;

(6) 实现 step 7 中的要求, 当 id=1 时, 遍历 openset, 将 openset 中与 neighborPtr 具有相同 index 的元素删除, 重新创建一个新的 GridNodePtr, 其属性与 neighborPtr 相同并加入到 openset 中;

(7) 实现 step 8 中的要求, 实现回溯, 从 terminatePtr 的 camefrom 中一步步回到起点。

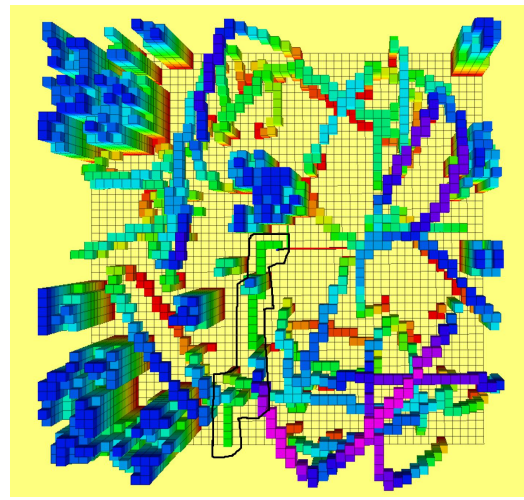
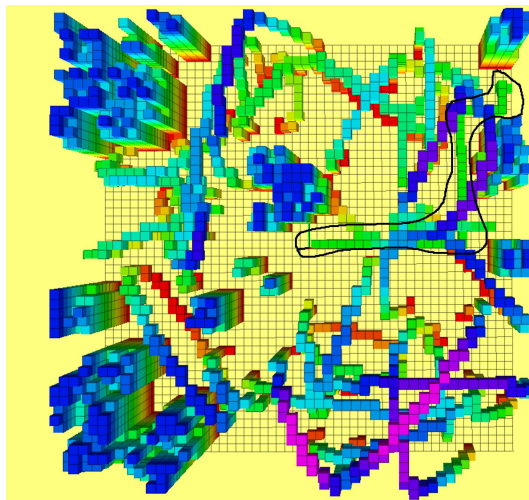
(二) 运行结果

(1) 欧式距离



```
[ INFO] [1574966088.956307252]: [node] receive the planning target
[ WARN] [1574966090.514047720]: [A*]{sucess} Time in A* is 1556.692194 ms, p
ath cost if 8.200000 m
count_back :41
[ WARN] [1574966090.515310119]: visited_nodes size : 32512
[ INFO] [1574966132.497129747]: [node] receive the planning target
[ WARN] [1574966134.550797609]: [A*]{sucess} Time in A* is 2053.575665 ms, p
ath cost if 9.000000 m
count_back :45
[ WARN] [1574966134.552490015]: visited_nodes size : 39193
[ INFO] [1574966144.715140675]: [node] receive the planning target
[ WARN] [1574966146.316727850]: [A*]{sucess} Time in A* is 1601.433054 ms, p
ath cost if 8.200000 m
count_back :41
[ WARN] [1574966146.317857213]: visited_nodes size : 32435
[ INFO] [1574966187.009444216]: [node] receive the planning target
[ WARN] [1574966189.358278880]: [A*]{sucess} Time in A* is 2348.723166 ms, p
ath cost if 9.600000 m
count_back :48
[ WARN] [1574966189.359837349]: visited_nodes size : 43689
```

(2) 曼哈顿距离



```
[ INFO] [1574965707.107885568]: [node] receive the planning target
[ WARN] [1574965708.674249388]: [A*]{sucess} Time in A* is 1565.224560 ms, path cost if 9.600000 m
count_back :48
[ WARN] [1574965708.675751253]: visited_nodes size : 36867
[ INFO] [1574965771.797246759]: [node] receive the planning target
[ WARN] [1574965772.581081999]: [A*]{sucess} Time in A* is 783.650848 ms, path cost if 7.600000 m
count_back :38
[ WARN] [1574965772.582108947]: visited_nodes size : 21618
[ INFO] [1574965816.253734484]: [node] receive the planning target
[ WARN] [1574965818.105948632]: [A*]{sucess} Time in A* is 1852.034928 ms, path cost if 10.400000 m
count_back :52
[ WARN] [1574965818.107591634]: visited_nodes size : 41769
```

(三) 启发函数对比

曼哈顿启发函数得到的路径由多根成直角的直线段形成，而欧式距离得到的路径是曲线；从所用时间和 path cost 来说，我未能看出明显区别。