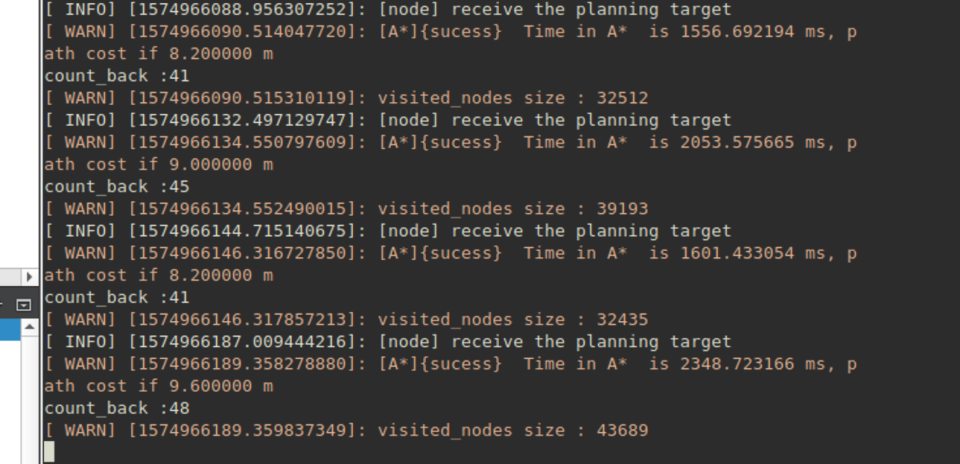
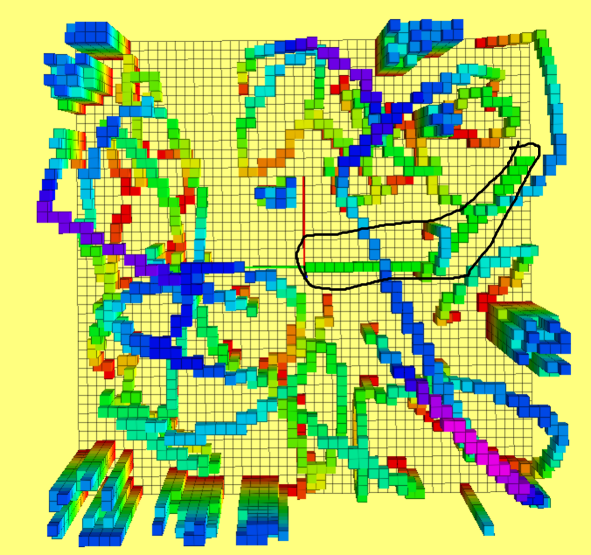
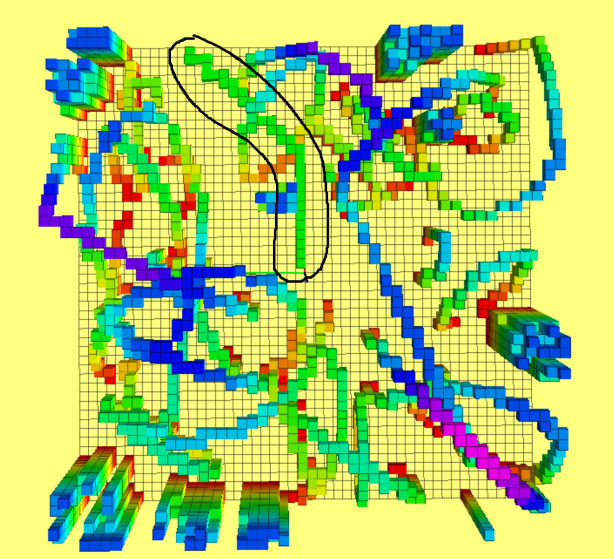
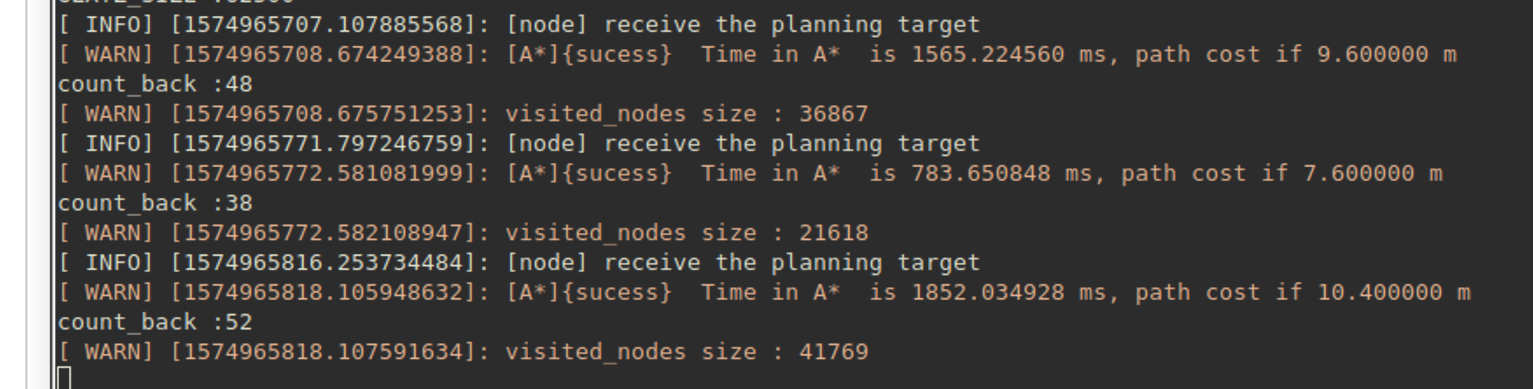
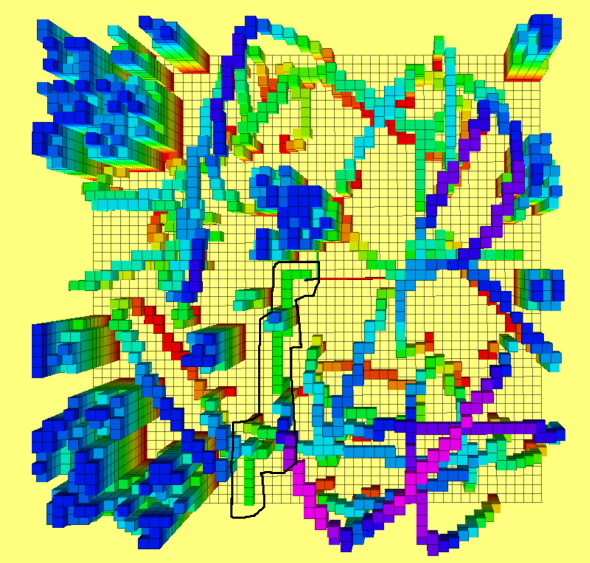
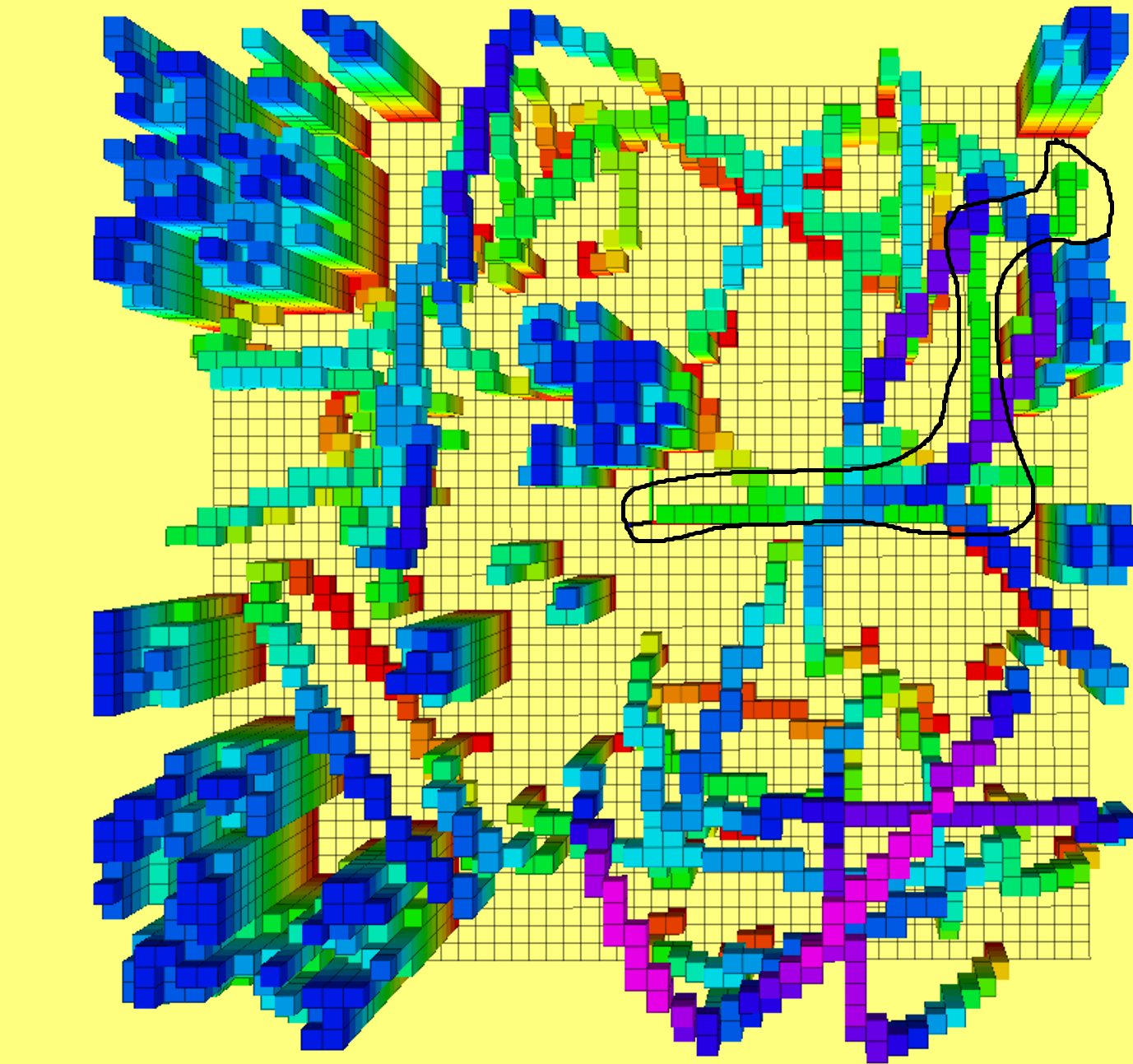
1. 算法流程：主要说明AstarGraphSearch函数中的实现
2. 实现step 1中的要求，即完成getHeu函数，在这个函数里接受两个指针参数GridNodePtr，分别对他们的坐标求欧式距离或者曼哈顿距离；
3. 实现step 2中的要求，把在初始化函数中已经初始化好的GridNodeMap中每一个node的fscore设置好；并把GridNodeMap中代表路径起点的node的id设置为1；
4. 实现step 3中的要求，将currentPtr设置为openset中的第一个，并把第一个从openset中删除；
5. 实现step 4 中的要求，即实现AstarGetSucc函数，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。
6. 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；
7. 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；
8. 实现step 8 中的要求，实现回溯，从terminatePtr的camefrom中一步步回到起点。
9. 运行结果
10. 欧式距离



1. 曼哈顿距离



1. 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。