代码说明：三个小问的代码写在一个工程中，使用CMakeLists.txt 里面的 #define来区分

add\_definitions(-DPROBLEM\_1)

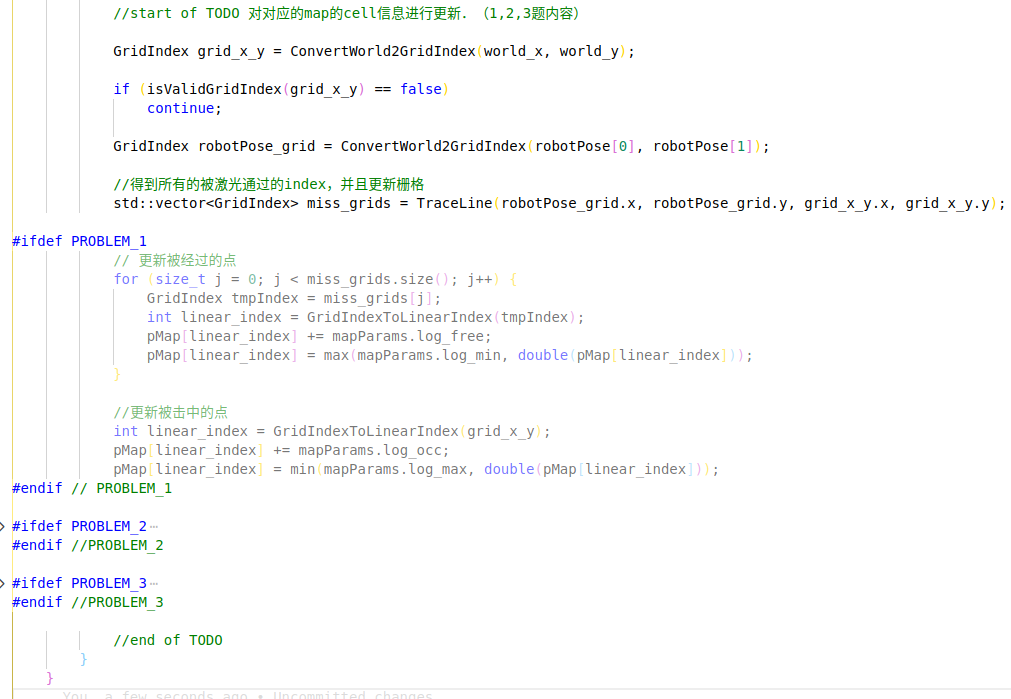
*# add\_definitions(-DPROBLEM\_2)*

*# add\_definitions(-DPROBLEM\_3)*

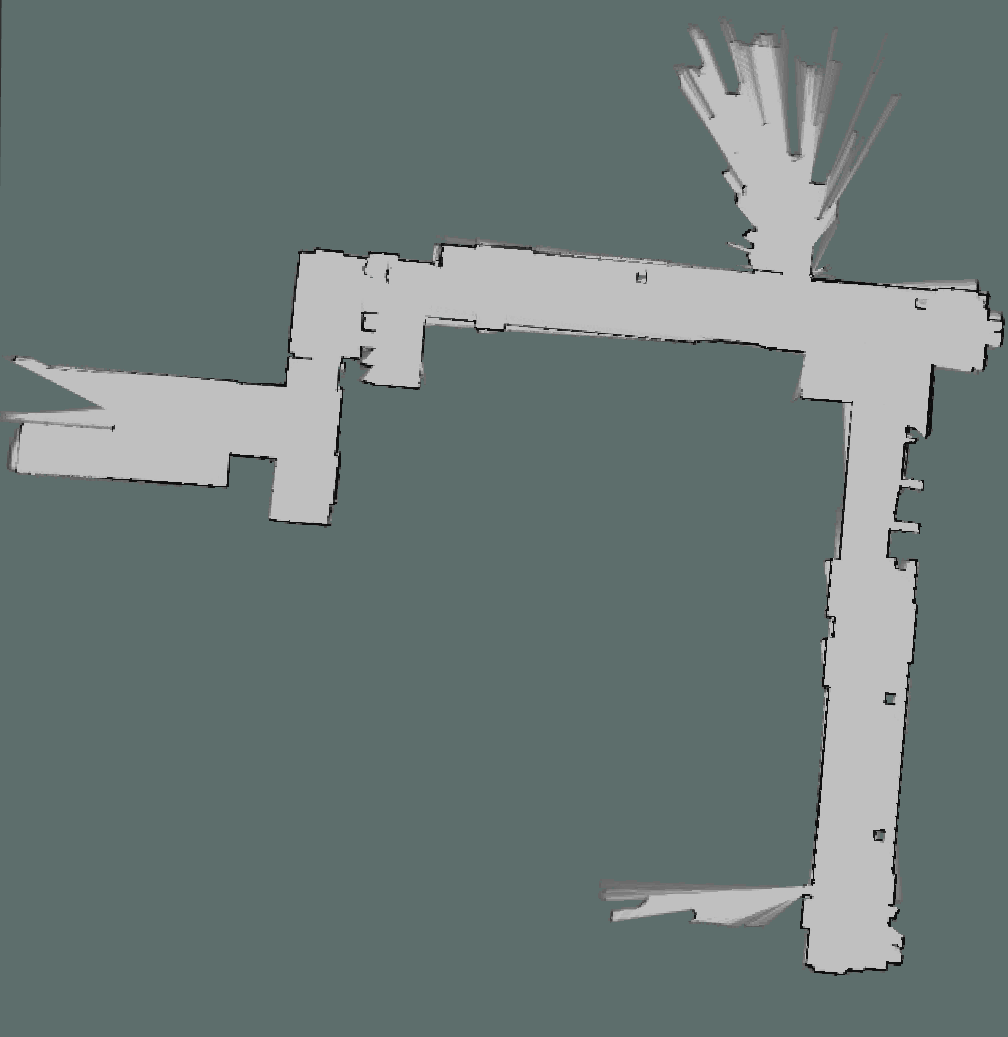
代码结构如下所示：



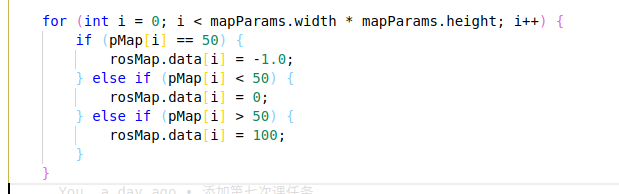
1. 补充代码，通过覆盖栅格建图算法进行栅格地图构建；（3 分）



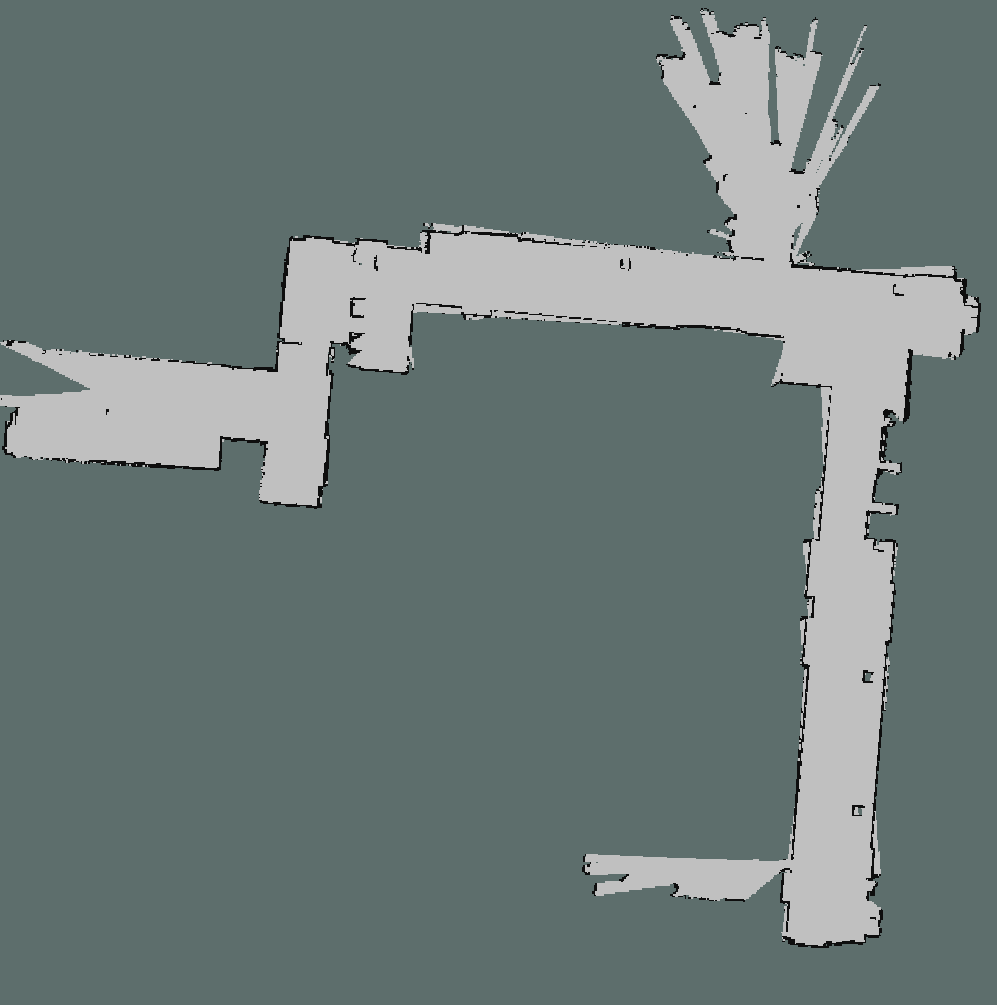
效果如下所示：



输出结果中间有些区域颜色比较淡，修改pub地图的函数如下所示，将pMap中小于50的栅格赋值为0, 大于50的栅格赋值为100.



输出结果如下所示，障碍物与非障碍物区域的对比效果更清晰。



不过这样做一些区域的障碍物被忽略掉了，比如左下角这里：

修改地图输出效果前：



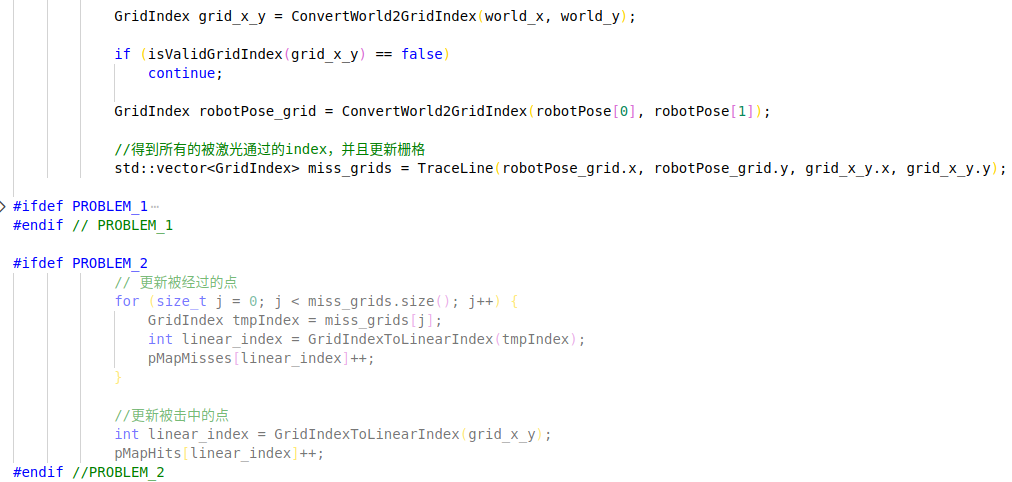
修改地图输出效果后：



1. 将第 1 题代码改为通过计数建图算法进行栅格地图构建；（3 ）

代码：

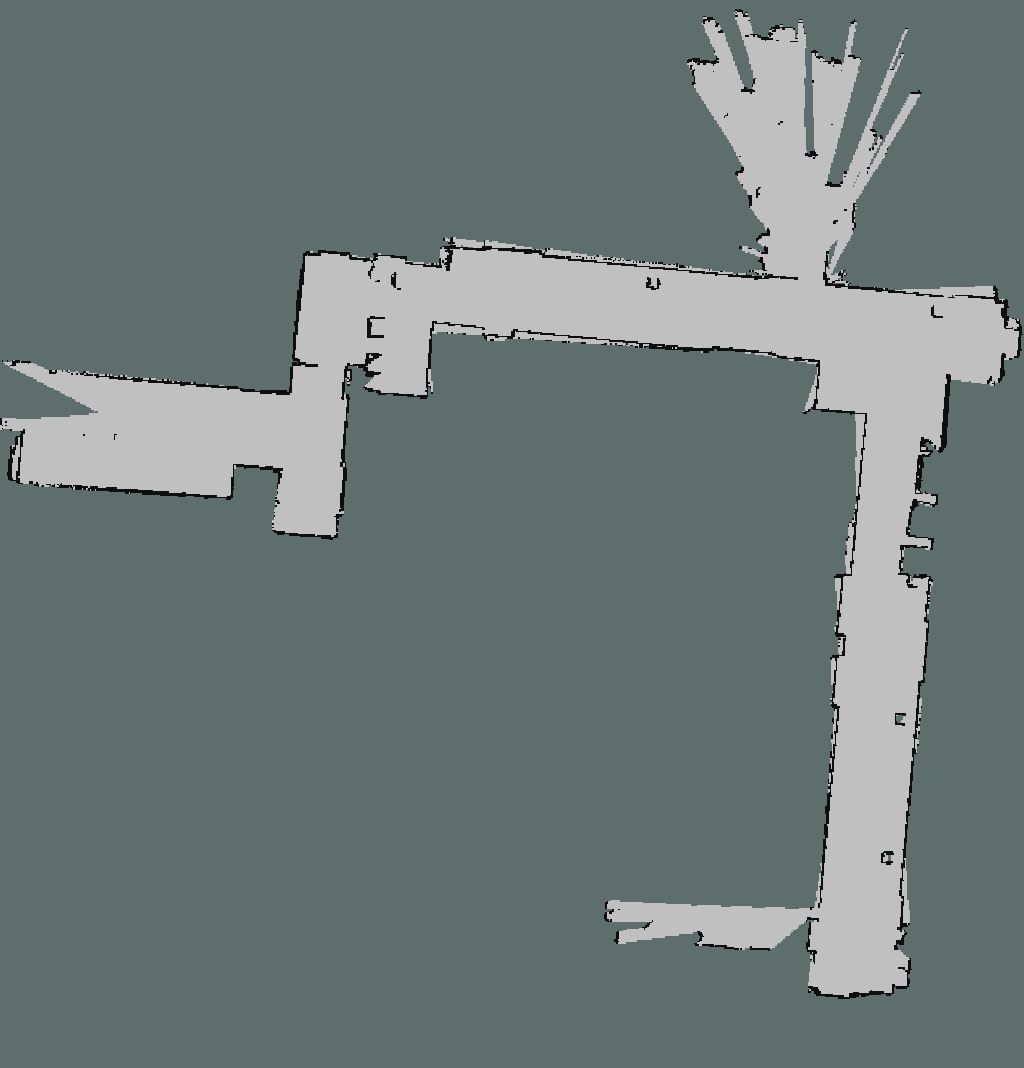
计数



计算比例，并赋值得pMap



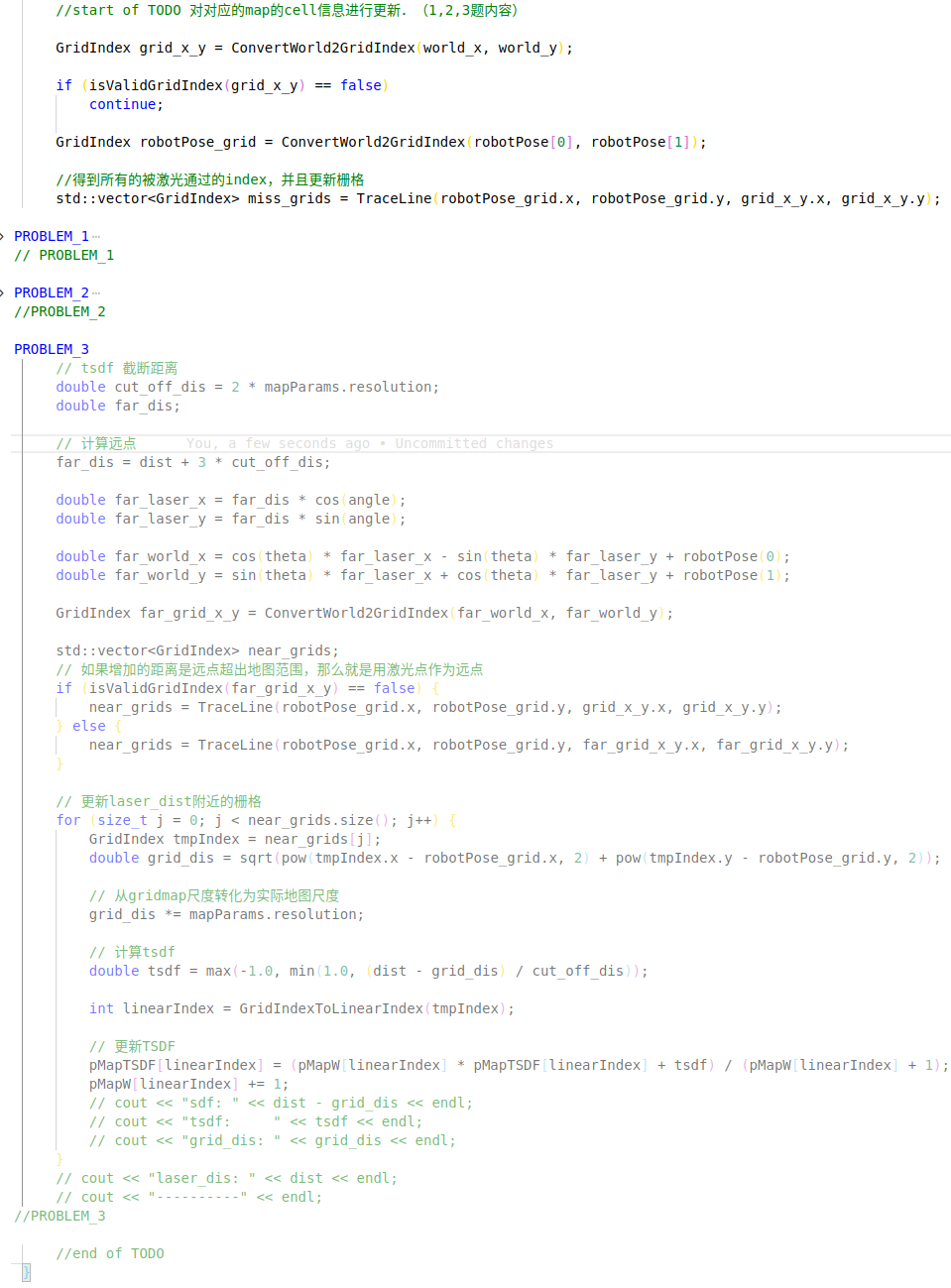
输出效果如下所示：



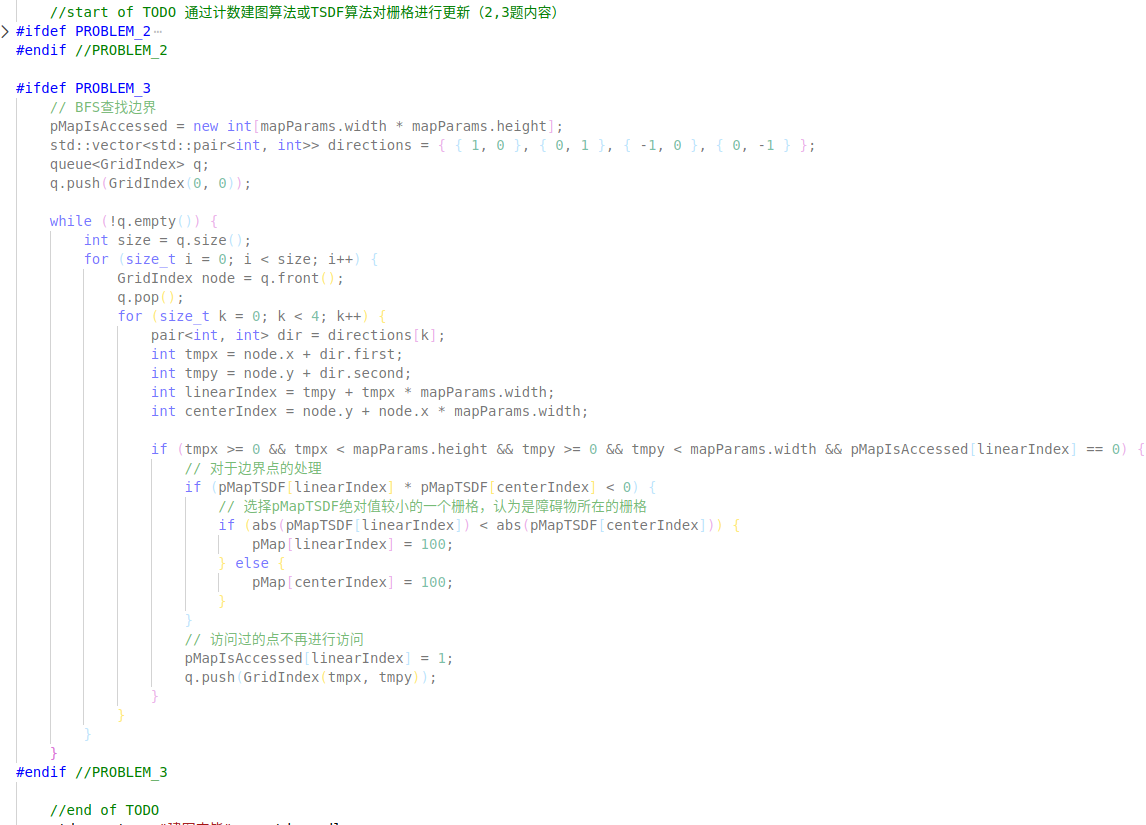
1. 将第 1 题代码改为通过 TSDF 建图算法进行栅格地图构建；（4 ）

代码：

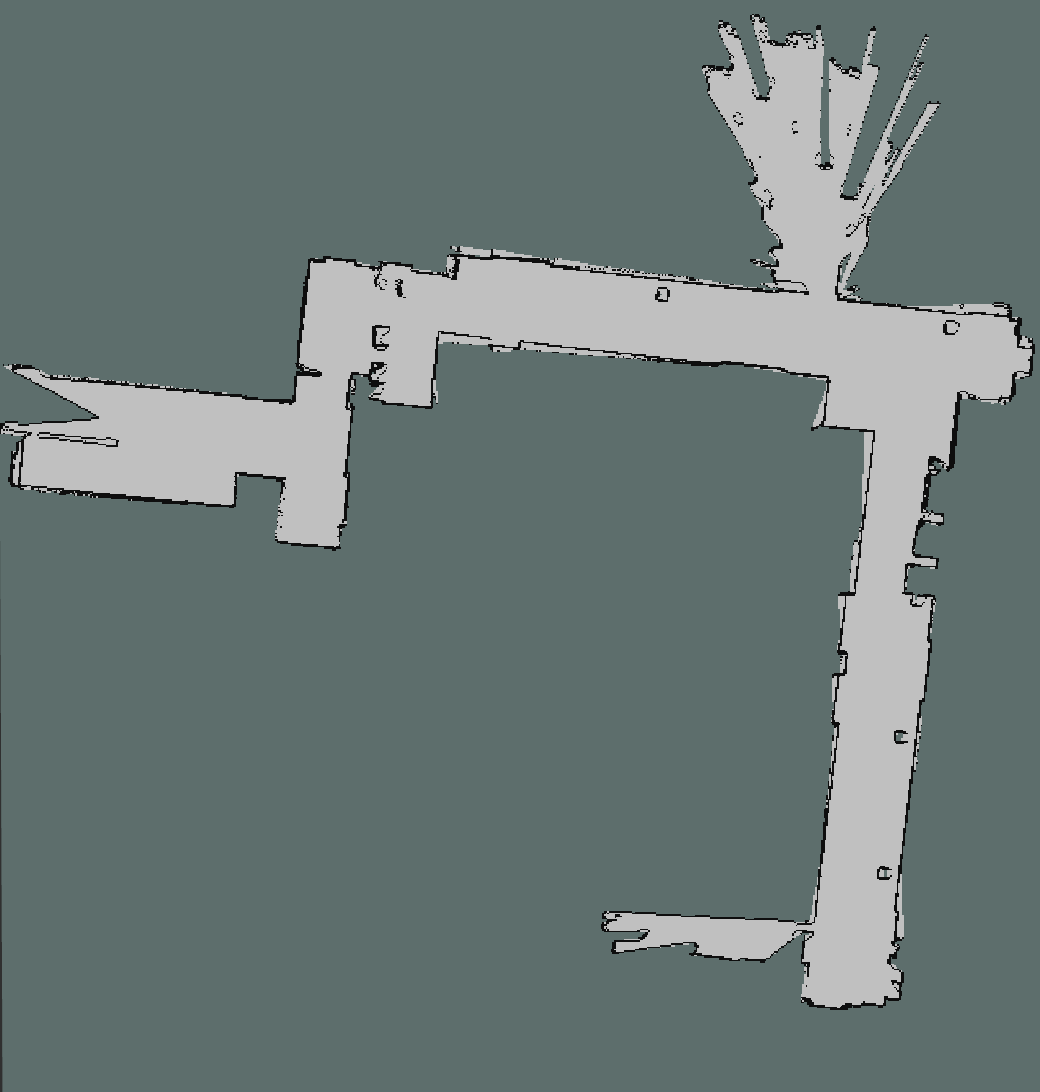
计算TSDF



广度优先搜索找到所有的边界处，并赋值对应pMap



输出效果如下所示：



1. 简答题，开放性答案：总结比较课堂所学的 3 种建图算法的优劣 （2 ）

1）覆盖栅格建圖算法对栅格进行更新只需要进行加法操作，具有较高的更新速度。

2）计数建圖算法实现简单，但是缺点是每个栅格需要两个计数变量Hits和Misses，内存占用较多。

3）上述两个方法没有考虑激光噪声，而TSDF建圖算法将多帧数据一起考虑

如果传感器的噪声服从高斯分布，那么通过TSDF进行融合，等价于通过最小二乘来进行融合，能比较好的进行曲面重构。

能够插值出确切的曲面位置，构建的地图最多只有一个栅格的厚度。不过该算法相比于前两种算法，计算复杂度较高