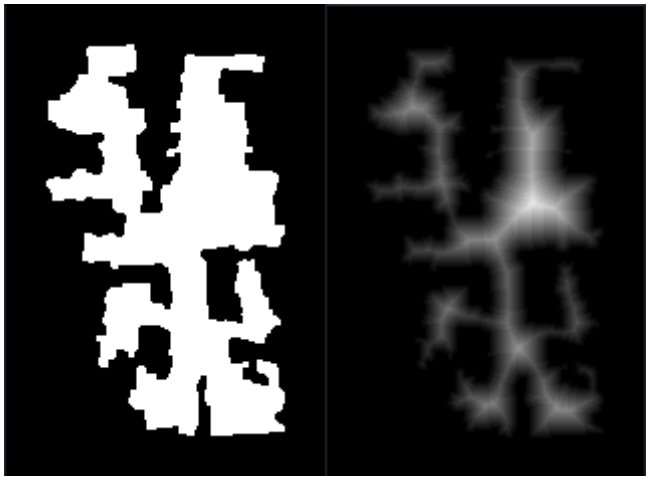
Room Segment Algorithm Base On Distance Map

Distance Map Description

Distance Map通过distance transform操作获得，在这个步骤中，前景（foreground）中的点会被转换为该点到最近的边界的距离，一个例子如下：



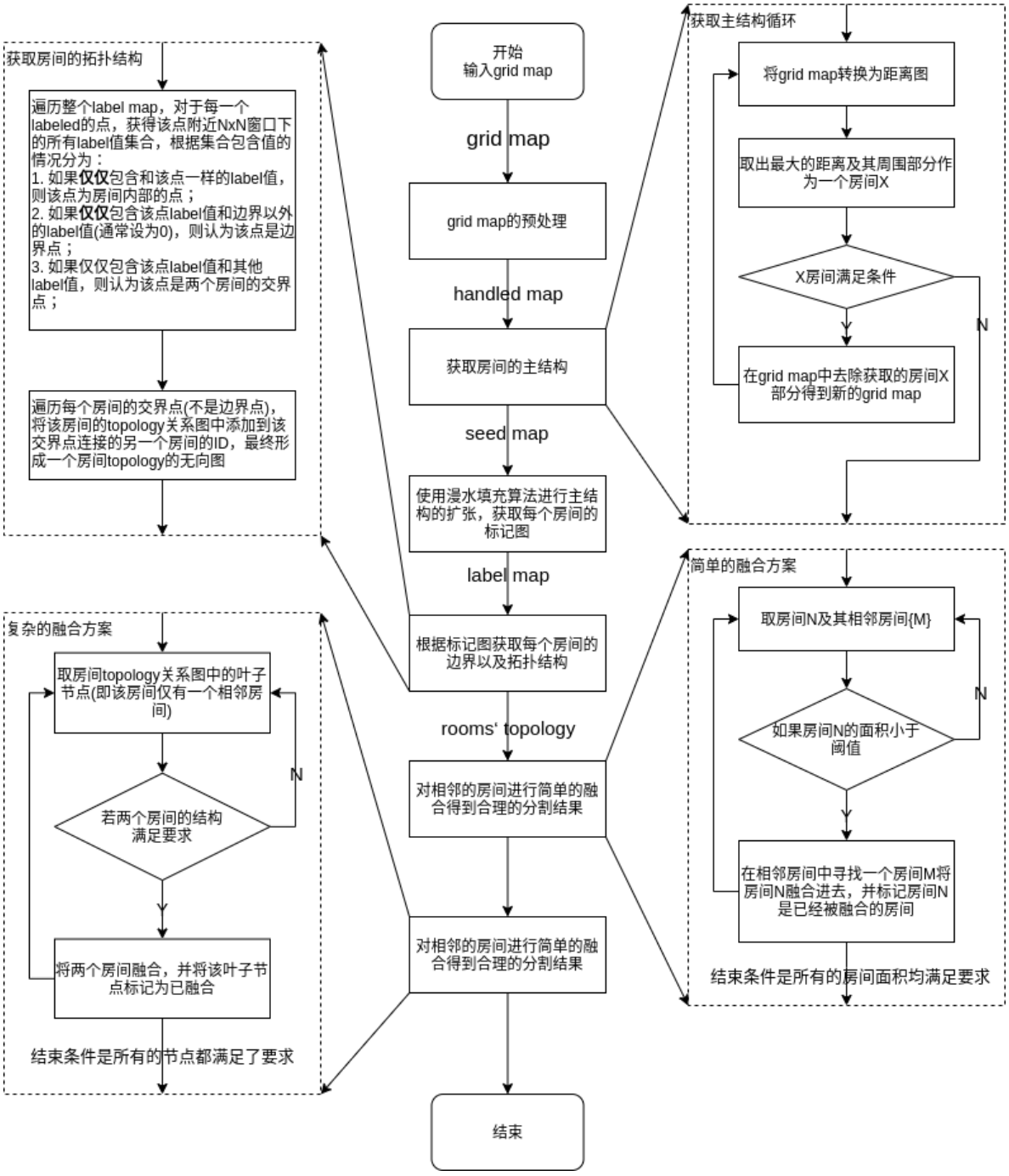
可以看到，如果把距离图看做一个等高线图的话，那么每个房间的中心的高度一定是很高的。

现有的论文如下：

Pedro Felzenszwalb and Daniel Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004.

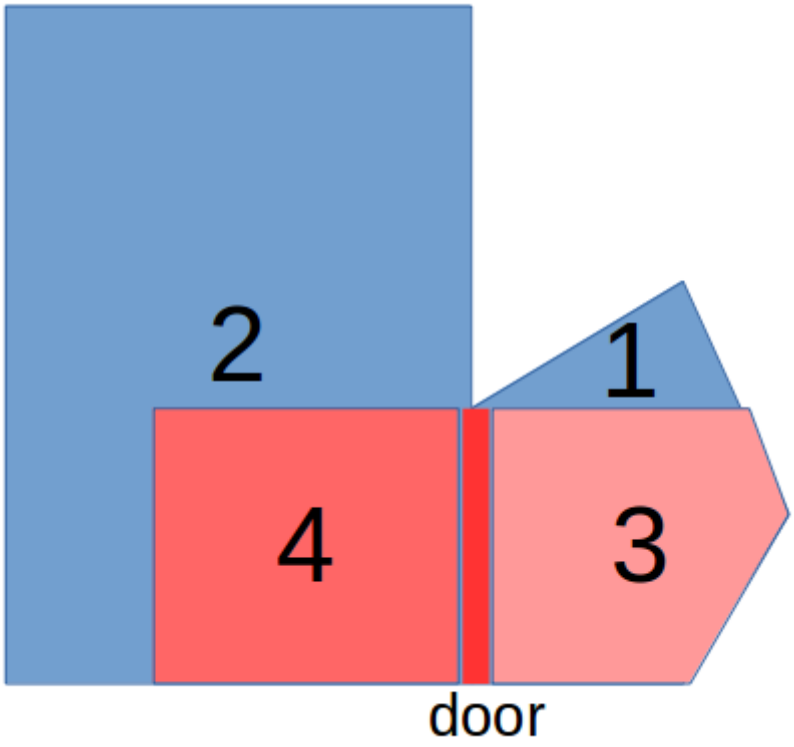
Main Algorithm Process

整个算法流程图如下：



所用的算法流程在上图中基本上都涉及了，这里简单说一下复杂融合方案中的融合条件；

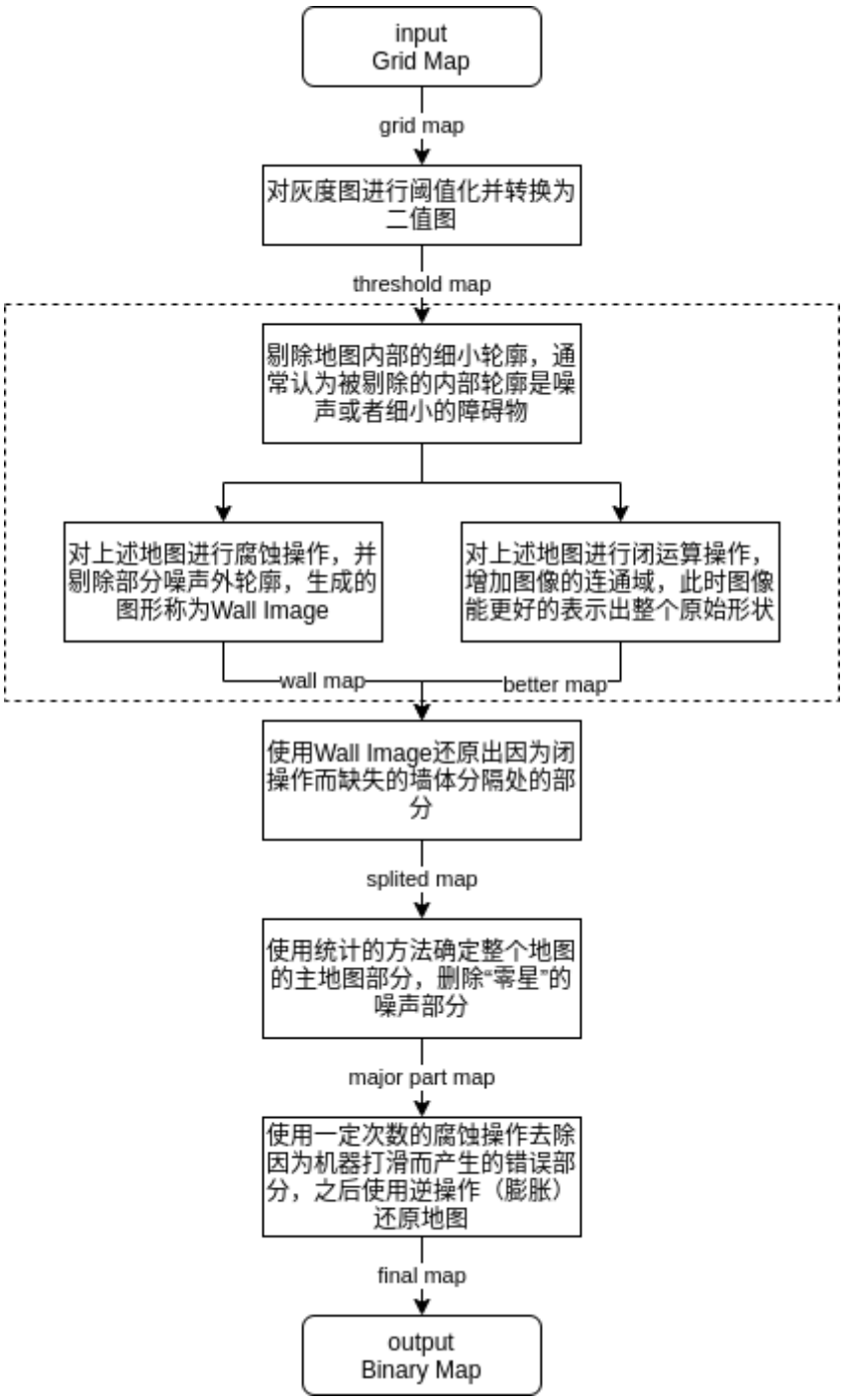
1. 如果两个区域可以称为是两个房间，那么它们中间的交界部分的长度需要满足现实世界的约束，这里选择为长度大于0.5m，之所以没有设计上限是因为有些时候两个“房间”之间的交界长度确实比较长，例如某些厨房和客厅之间的分界处的长度就较大；
2. 如果两个区域可以称为是房间，那么沿着交界线的方向进行一定长度的垂直扩充，则扩充区域与该区域的IOU必定小于一个阈值，如果大于该阈值，则倾向于认为该区域是误分的，需要被合并，该过程可以表示为如下过程：



其中3表示扩充区域与原区域1的重叠区域，4表示扩充区域与原区域2的重叠区域，可以看到，扩充区域有一定的范围，且该扩充区域的边界和原区域的边界是重合的。所以该例子下，因为area(3)/area(1)接近1，所以认为1这个区域是误分的，会和区域2进行合并。

Grid Map预处理环节

预处理阶段主要通过阈值化、图像形态学处理等操作，尽可能的还原机器人给出的地图原先的模样，整个预处理过程如下：

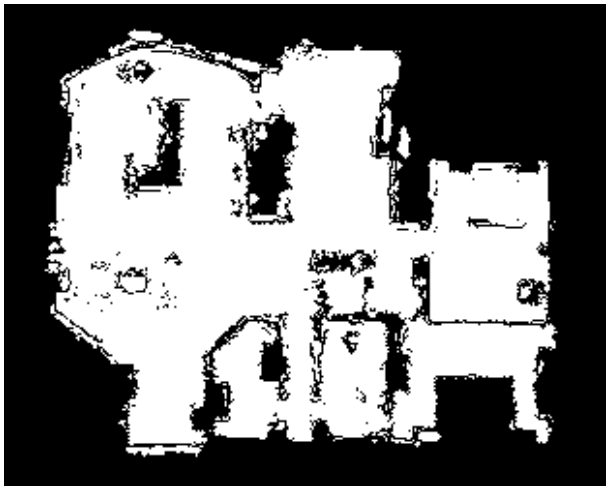


下面是一个实际的例子：

输入的GridMap

对灰度图进行阈值化

下图为流程图中的threshold map。



对阈值化之后的图进行初步的处理

包括去除细小的障碍物，生成wall image和形态学处理之后的图像。

左图为经过了后续操作之后的Wall Image，右图为形态学处理之后的Better Image。

可以看到，wall image基本上把所有的墙体部分都完好的分割了出来。

使用Wall Image还原墙体分割部分

可以看到下图和上图中右侧的部分相比，墙体部分被很好的还原了出来，而其他部分则基本不受影响。

下图为流程图中的splited map。

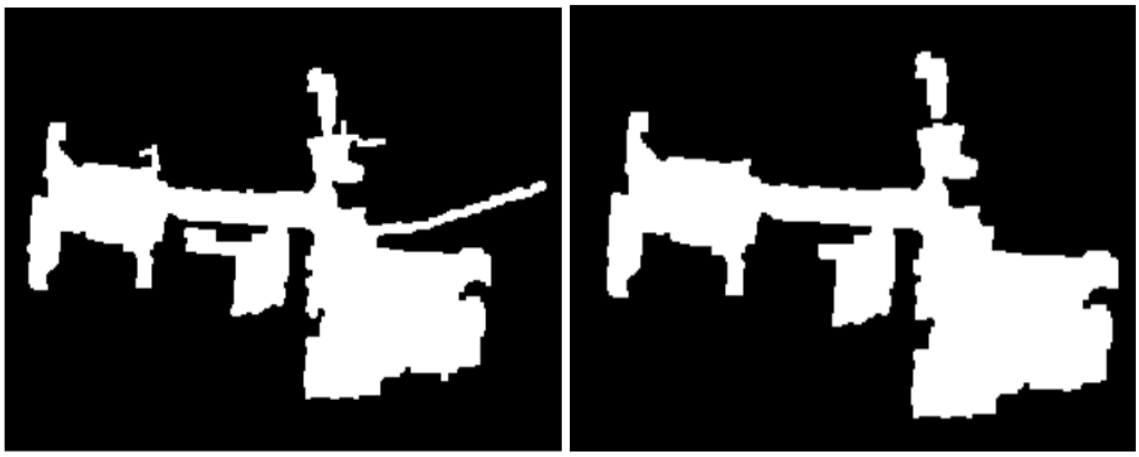
使用统计方法获取主地图部分

与上图对比之后可以发现，红框所包含的区域属于噪声区域，被算法滤除了，而绿色框包含的区域属于整个地图的主部分，虽然其连接处是断开的，但是算法进行了保留。

去除打滑区域

由于该例子中不存在打滑区域，因此这个地方换一个例子说明。

可以看到，机器人打滑的区域被算法去除掉了，同事其他部分被较好的保留下来



漫水填充算法

关于漫水填充算法，现有的论文如下：

Fernand Meyer. Color image segmentation. In *Image Processing and its Applications, 1992., International Conference on*, pages 303–306. IET, 1992.

因为本算法针对的是二值图(仅有0和255)，所以算法上稍微简单很多，算法伪码如下：

```
temp_image = input_image

finished = False
while (False == finished):
    finished = True
    for x = 1; x < w-1; x++:
    for y = 1; y < h-1; y++:
        if (spreading_map[x, y] == 255):
            set_value = False
            for dx = -1; dx <= 1 and set_value == False; dx++:
            for dy = -1; dy <= 1 and set_value == False; dy++:
```

```
                label = input_image[x + dx, y + dy]
                if (label != 0 && label < 255) {
                    temp_image[x, y] = label
                    set_value = True
                    finished  = False

    image = temp_image
```

获取房间主结构

整个获取房间主结构的过程如下面例子所描述：可以看到每个房间的主结构被一个个的提取出来。

漫水填充获取房间的标记图

进行不同房间融合

相比于上面的label-map，房间7被融合进入了房间1中，房间与房间之间的top关系如图中绿色线段所示。