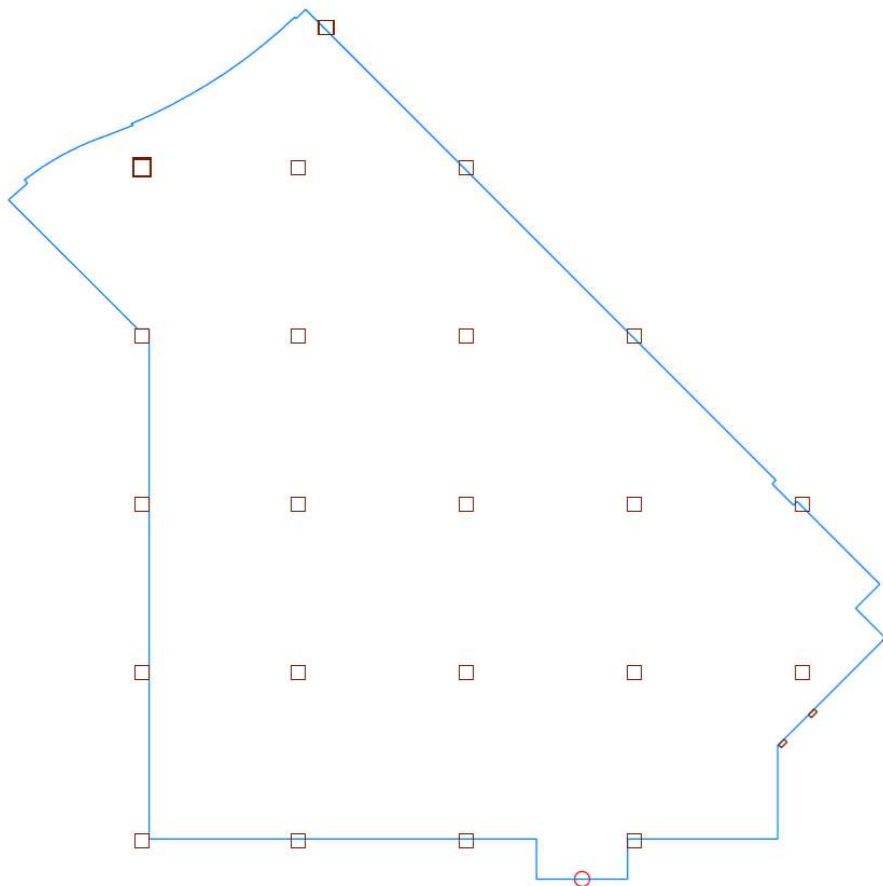


复杂场景的疏散路径生成-柱网法

输入条件

输入为 Geojson 图形，Geojson 数据如下图所示：



附图 1 输入条件示例-平面信息

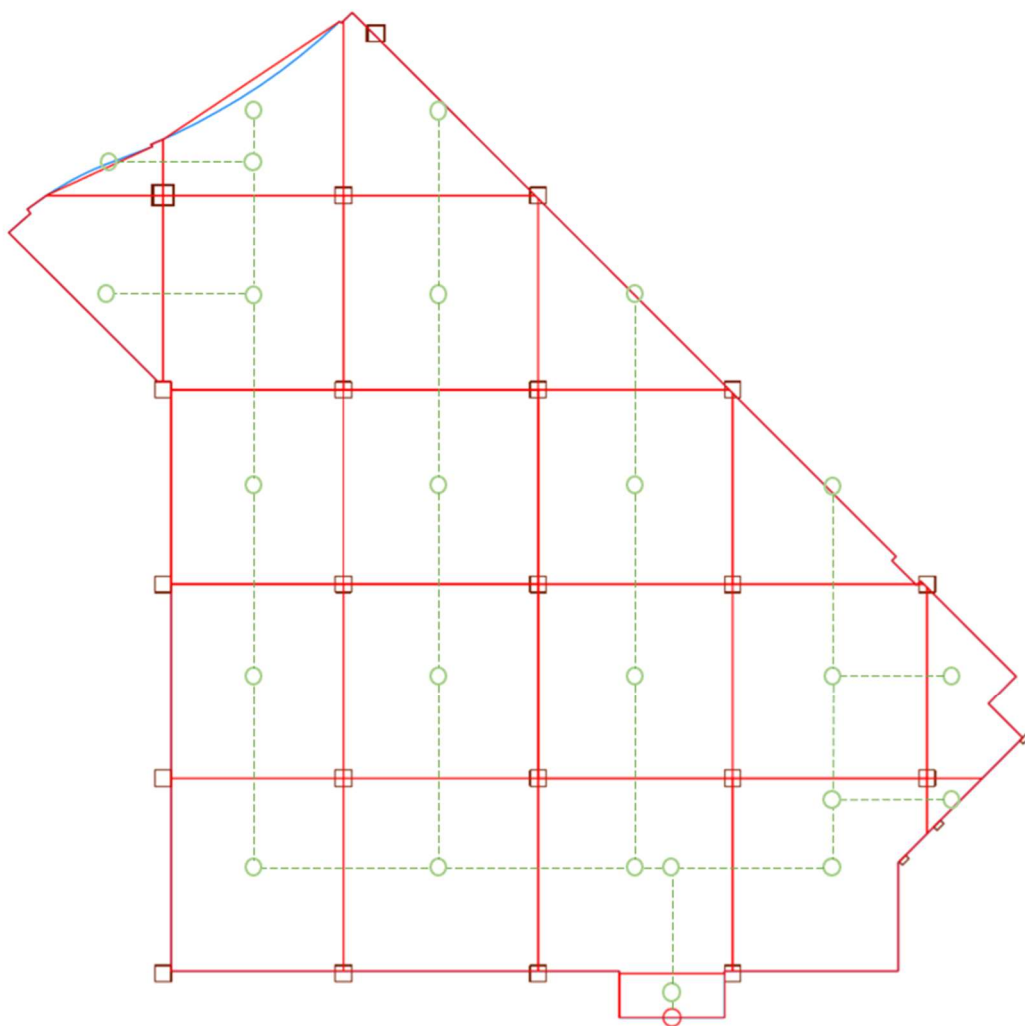
1. 蓝色：房间轮廓线(Polygon)，由墙、门、窗、卷帘、柱、剪力墙围合形成；
2. 黑色：柱、剪力墙等障碍物(Polygon)；
3. 红色：出口位置(Point)；

处理要求

疏散路径横平竖直

输出结果

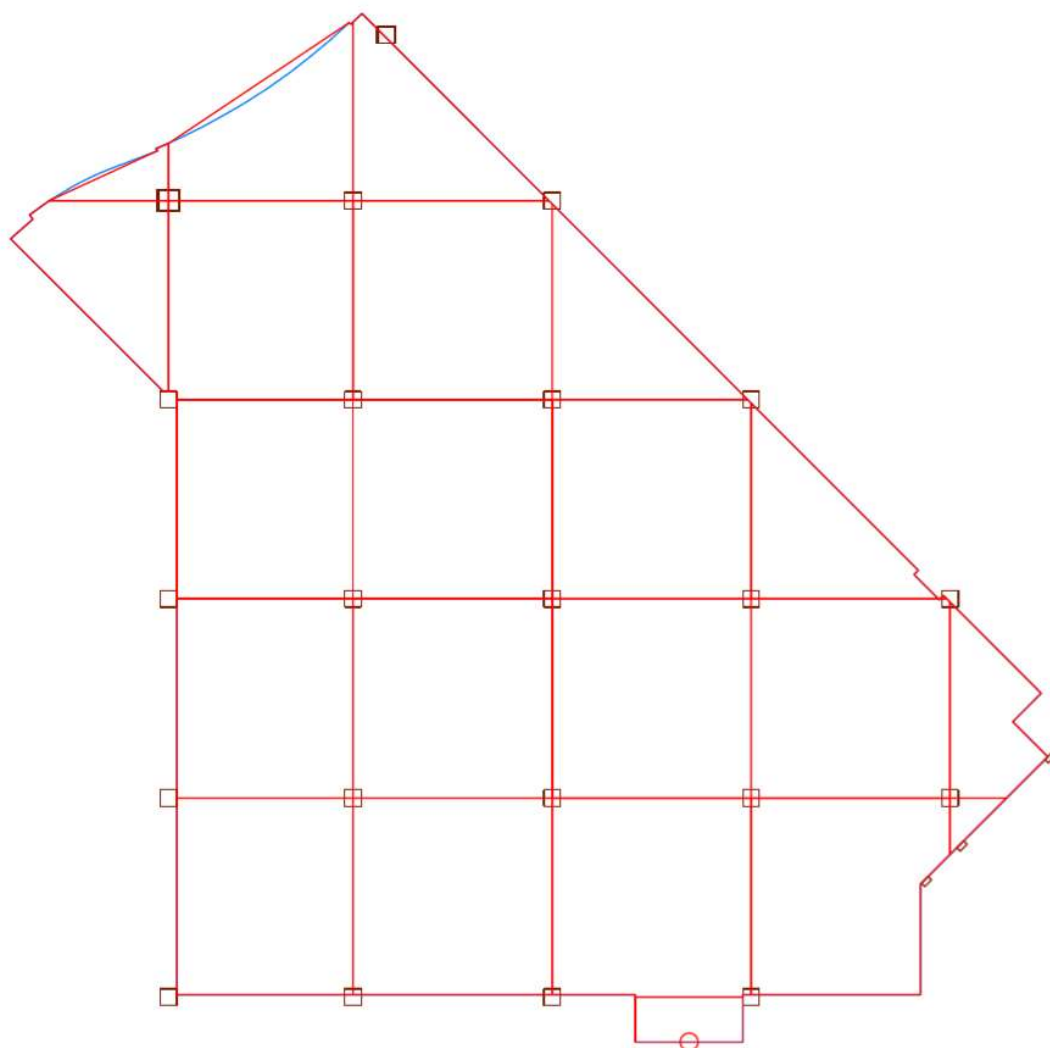
疏散路径（下图绿色虚线）



附图 2 输出结果-疏散路径

疏散路径作为中间结果，用户确认后作为布置疏散指示灯的输入条件。

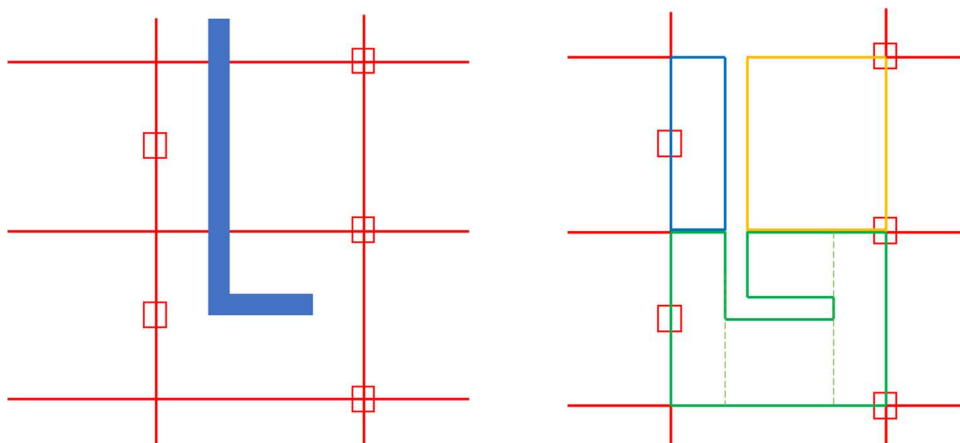
实现思路



附图 3 矩形区域生成柱网

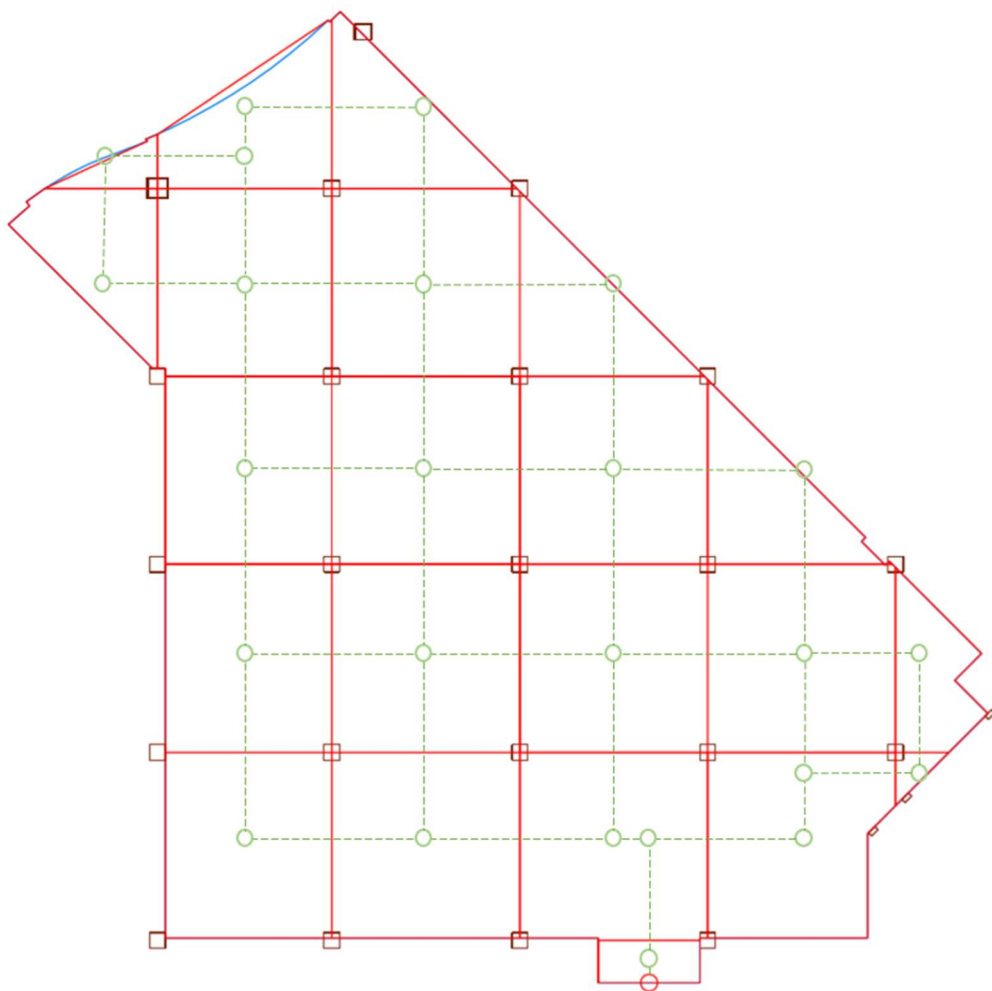
1 生成柱网

- (1) 使用已有算法生成横平竖直的网格。要将房间轮廓线外扩一圈后，获取内部的所有结构柱、剪力墙，根据柱的位置形成柱网。
- (2) 对于包含剪力墙的网格，去掉剪力墙内的部分
- (3) 在房间轮廓线上的网格，去掉轮廓线外的部分
- (4) 减去形状后，若形成凹多边形网格，则进行拆分（在大于 180 度的内角处，引一条横平竖直的线到边界。极端情况仍待考虑。是否不需要进行拆分？）



附图 4 左图蓝色多边形为柱网中的剪力墙，右图是减去剪力墙后形成的柱网

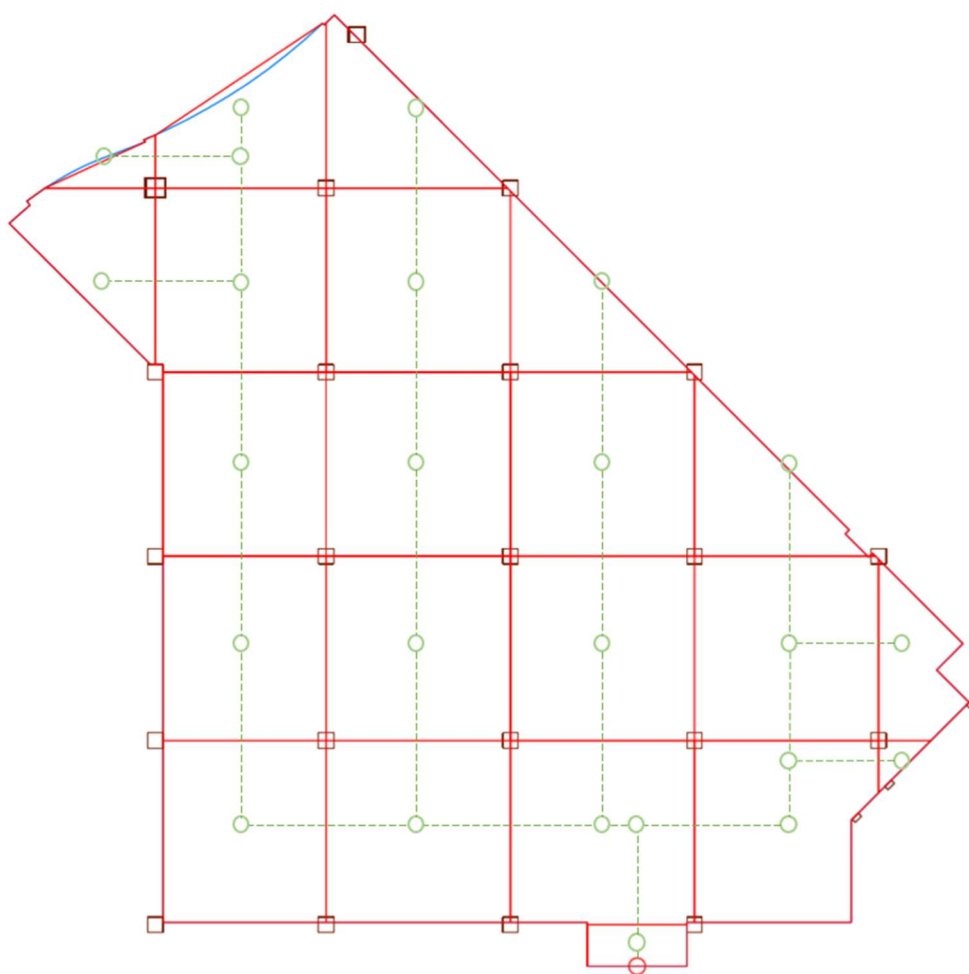
2 生成疏散网格



附图 5 疏散网格

- (1) 在每个柱网内取一点作为疏散点。目前想法是取横平竖直的外接矩形的中心点作为疏散点，若不在柱网内部，则进行移动。
- (2) 连接相邻柱网的中心点形成疏散网格。采用曼哈顿连接使线路横平竖直，若遇障碍采用 A* 寻路（保证所有疏散线路不穿障）。在弯折处形成新的疏散点（保证疏散点到疏散点的路径都是一条横平或竖直的线段）。
- (3) 疏散出口所在的柱网内，将疏散出口连接到最近的疏散点
- (4) 将重合的路径合并，形成最终的疏散网格。

3 生成疏散路径



附图 6 疏散路径

采用 Dijkstra 算法，以树的结构保存疏散路径

- (1) 设一个虚空根节点，记为原点
- (2) 将所有疏散出口作为原点的孩子，并根据 Dijkstra 算法，对可达疏散点做更新。
- (3) 每个疏散点加入树后，考察可达疏散点，若距离更短则更新，若距离相同则判断到出口的转弯次数，更少则更新（保证距离最短，转弯最少）
- (4) 同一个柱网内的疏散点不应被分散到多条路径上，故当某个疏散点加入树后，立即先将同一个柱网内的其他所有疏散点添加到该疏散点为根的子树中，再对其他柱网继续考察（保证一个柱网内不会有多个方向）
- (5) 去掉原点，得到各疏散出口为根的森林，得到最终的疏散路径（各疏散点在入树时就分配好了的出口）