

实验八 无人机 SLAM 基础 – 立体环境建模与路径规划

1.1 实验目的

在无人机的各种应用中，测绘作为一种重要功能，已经在近年来得到广大研究者关注。激光雷达作为获取目标散射点三维信息的工具，是无人机在测绘领域应用的重要传感器。

本次实验的目的是在 PCL 库的基础上掌握 C++ 的基本编程，以及基本点云分析、地图构建方法的实现。在前序 PCL 库配置实验的基础上，充分利用 PCL 库以及 Visual Studio 强大的功能对现有的点云数据进行环境建模分析，并利用建模的环境信息进行简单的路径规划。实现基本的障碍物规避，路径优化以及算法比较等任务。了解基本的图论算法在路径规划中的应用，并使用 C++ 和 PCL 库对算法进行实现。

1.2 实验原理

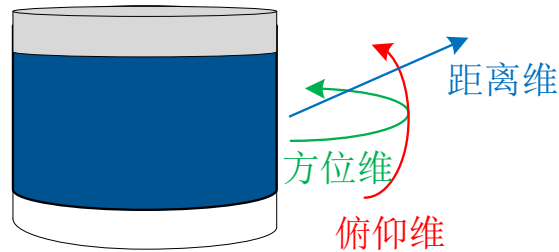
1.2.1 激光雷达探测原理

本次实验的主要任务是利用激光雷达获取的环境信息，根据环境中的散射点分布，建立合适的模型以表示障碍物和自由空间的相对位置关系。并在此基础上，给定起始点和终点，设计路径搜索算法实现室内场景下无人机的航迹规划。

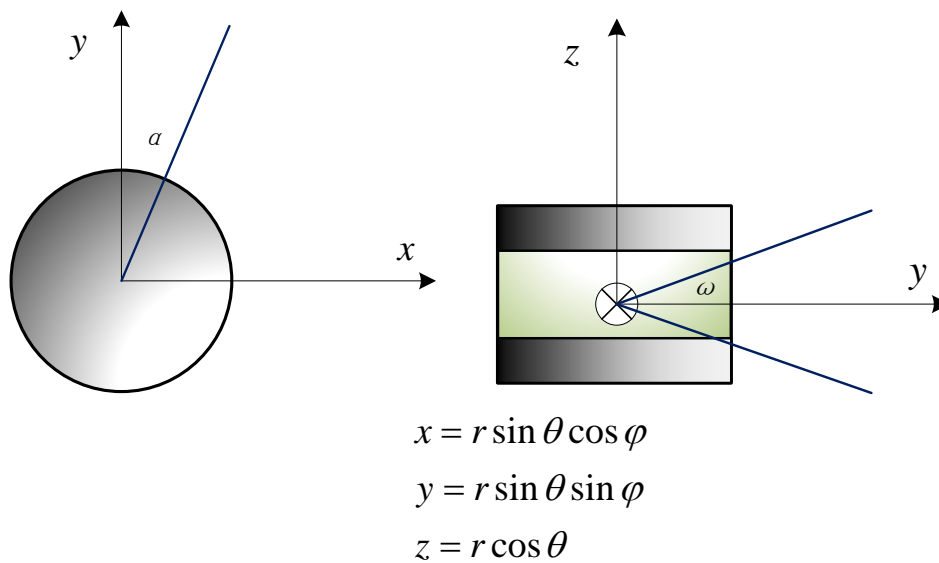
激光雷达作为获取环境信息的重要传感器，其成像是对于环境场景进行三维模型建立的最直接的方式。激光雷达的成像原理与雷达类似，它利用光的传播特性，发射激光脉冲并统计回波的峰值时刻，通过其与波长的比对，从而获取目标的距离信息。



如下图所示为 Velodyne-VLP16 型号 16 线激光雷达的成像原理示意图。可以看出，它利用极坐标系与直角坐标系之间的坐标转换关系，通过在指定的方位，指定的俯仰角度下获取散射点的距离信息，从而确定散射点在三维坐标系下的 x,y,z 信息。



根据直角坐标系与极坐标系的坐标关系可知



当激光雷达高速旋转时，其会以一定的重复频率发射脉冲。同时，系统的存储器记录了发射脉冲时刻用于区分不同脉冲，此外，系统还会记录该时刻视线方向所对应的水平方位角度和垂直俯仰角度。通过对该时刻发射脉冲回波距离维信息进行分析，结合上式即可得到空间中，探测距离范围内所有散射点的回波坐标信息。

可以看出，激光雷达的距离分辨率取决于脉冲宽度，角度分辨率则取决于控制方位维和俯仰维电机的最小旋转角度。根据俯仰维的角度分辨率，激光雷达可以分为 16, 32, 64 线激光雷达。相应地，其俯仰角度分辨率逐渐提高，系统成本也随之逐渐增加。

1.2.2 路径规划问题综述

路径规划是指给定起始点和目标点，以及环境信息，对无人机飞行航线进行设计和优化的问题。其主要包括两大部分：环境建模以及路径设计。

(1) 环境建模

根据激光雷达的成像原理可知，激光雷达的分辨率取决于脉冲宽度，方位角

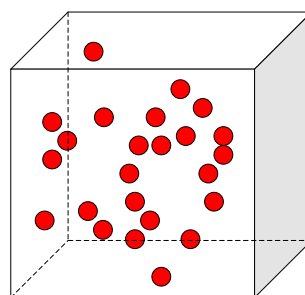
分辨率以及俯仰维角分辨率（线数）。商用的激光雷达技术比较成熟，可以在一定的探测范围内通过高重频脉冲获得较高的分辨率，最终将散射点的信息反映到成像系统中，生成点云。

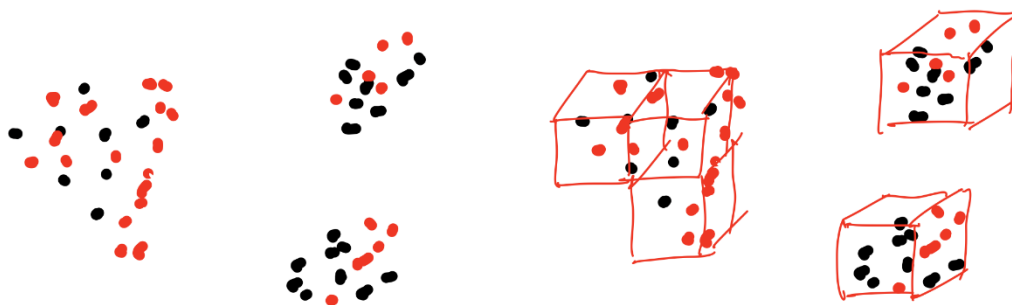
然而，激光雷达的高分辨特性也有其不足之处 – 即高分辨率势必会带来巨大的数据量和不必要的噪点。如下图所示为 12 米长，10 米宽，4 米高的某实验室中由 16 线激光雷达在 10ms 内录取点云信息数据量大小。共 27028 个点，每个点包括 xyz 信息，大小为 1MB。

test2.txt - 记事本		
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)		
0.009653132	1.675992846	-0.269276291
0.022361765	3.768293142	-0.468443692
0.023632791	3.761225939	-0.20521462
0.024975644	3.765731812	-0.075733036
0.033650015	4.943523407	1.301286221
0.034539931	4.947400093	1.120372176
0.035311785	4.934589386	0.940215468
0.03625112	4.945235252	0.76469785
0.037071612	4.939550877	0.589471102
0.037995078	4.947533607	0.417683154
0.038716644	4.929454803	0.246411487
0.030153301	3.755695105	0.053808682
0.014830451	1.666078091	-0.267710239

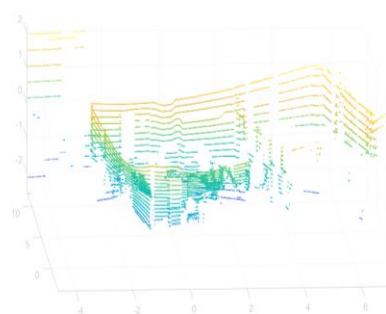
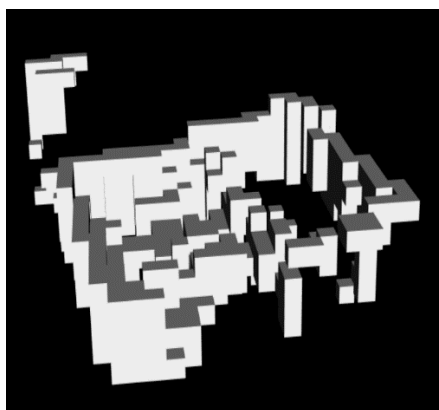
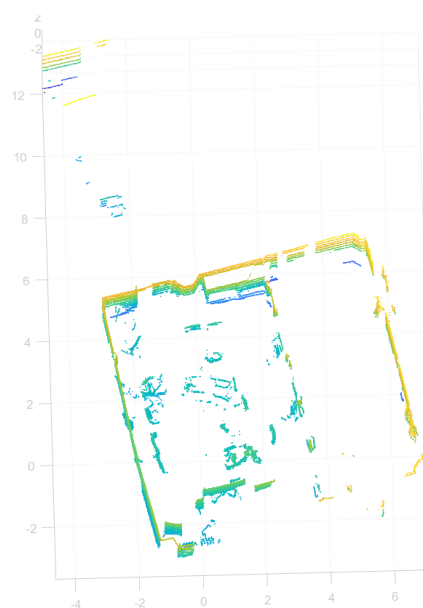
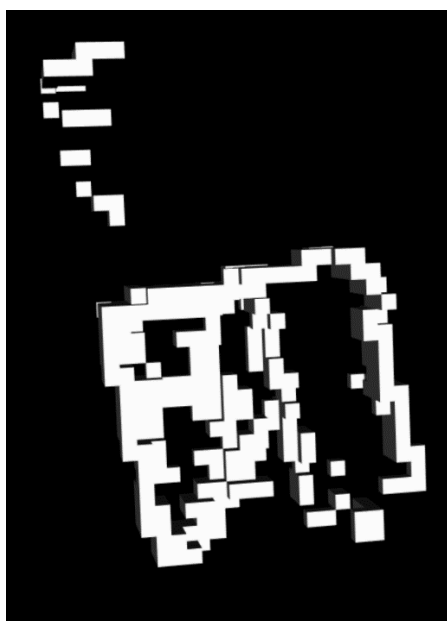
若考虑无人机系统对于路径规划的实时性要求，在 10ms 内处理 1MB 的数据则至少需要 100MBps 的带宽传输数据。此外，算法本身还具有一定的复杂度和运行时间也需要考虑进来。受自重影响，芯片功耗限制要求也比较高，无人机本身具有一定体积，安全飞行需要与障碍物保证一定距离……综上考虑，在路径规划之前，有必要对环境模型进行离散化的建模。

一种比较简单的环境建模方法就是根据点云信息，将环境分成若干个正方体作为网格单元。如果该网格单元中存在物体的散射点，整个立方体就视为障碍点；反之，该立方体就是可以飞行的自由空间。基于立方体的环境建模方法如下图所示。





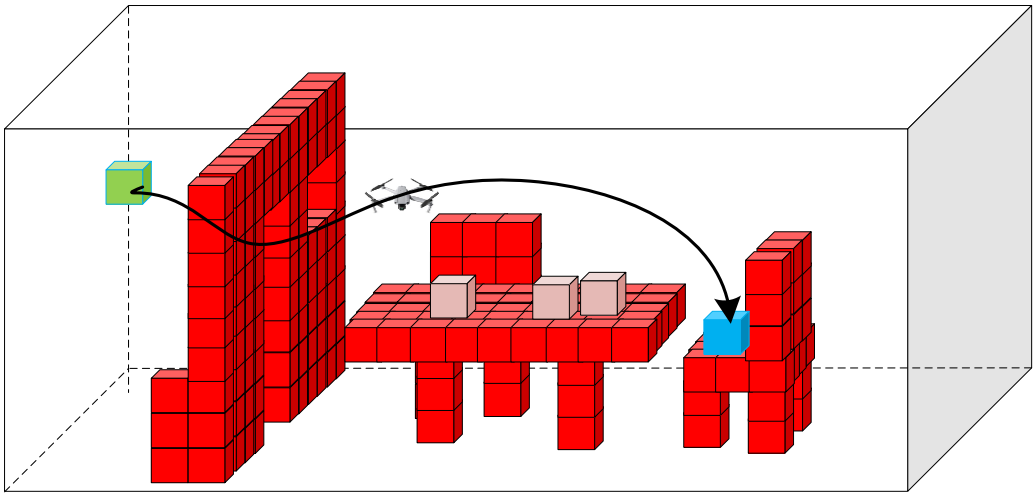
其中，红色方框表示该点为障碍物点，灰色方框表示该方框为自由空间，无人机可以在该空间中飞行。按照这种方法对于某实验室的环境模型建立效果如图所示。左图为按照上述原则建立的环境障碍物模型，右图为原始点云模型



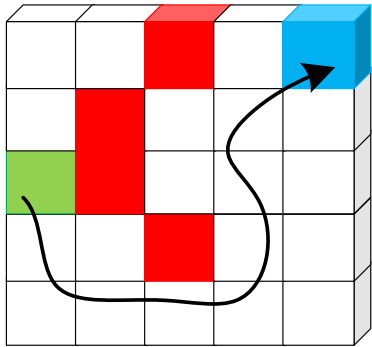
(2) 路线设计

环境模型在考虑了无人机飞行的实际问题基础上，对环境进行了建模，使用了空间采样的方法减少了路径规划的计算量。接下来，还需要根据起始点、终点设计出规避障碍物的路径规划算法。

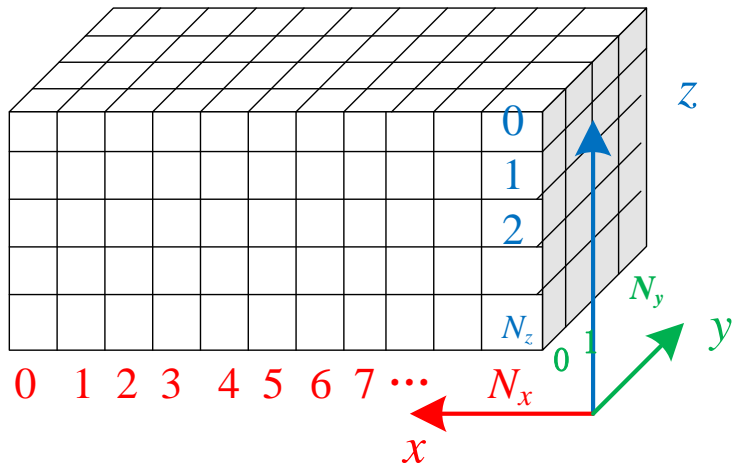
如下图所示为障碍物规避路径规划的示意图。绿色块代表起点，蓝色块代表终点，红色块代表障碍物，蓝色曲线代表路线。



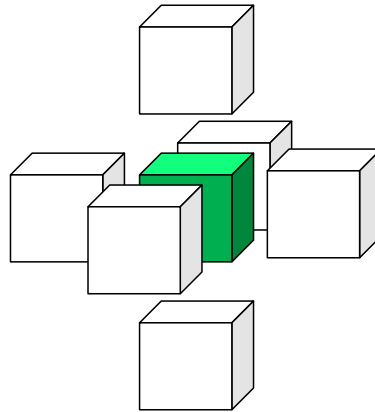
简化的二维路径规划问题则如下图所示



为了方便表示经过的路线，则可以将各个网格单元按照 x,y,z 的维度顺序进行标号。标记出该位置在三维各个方向的索引，以及在全局所有点中的位置，如下图所示。



这样，每一个网格都有自己在全局中的标号，以及其在三个方向的索引。二者可以互相计算，互相转换。接着，对该点的邻域进行分析，如下所示



根据一定的准则，在每一步选择合适的邻居点作为下一步的位置，即可完成路径规划。

1.2.3 路径规划方法指标分析与评价方法

路径规划问题是一个比较复杂的问题。由于其本质是基于环境信息的搜索，因此需要将环境信息的转换成搜索的条件加到判定过程中。

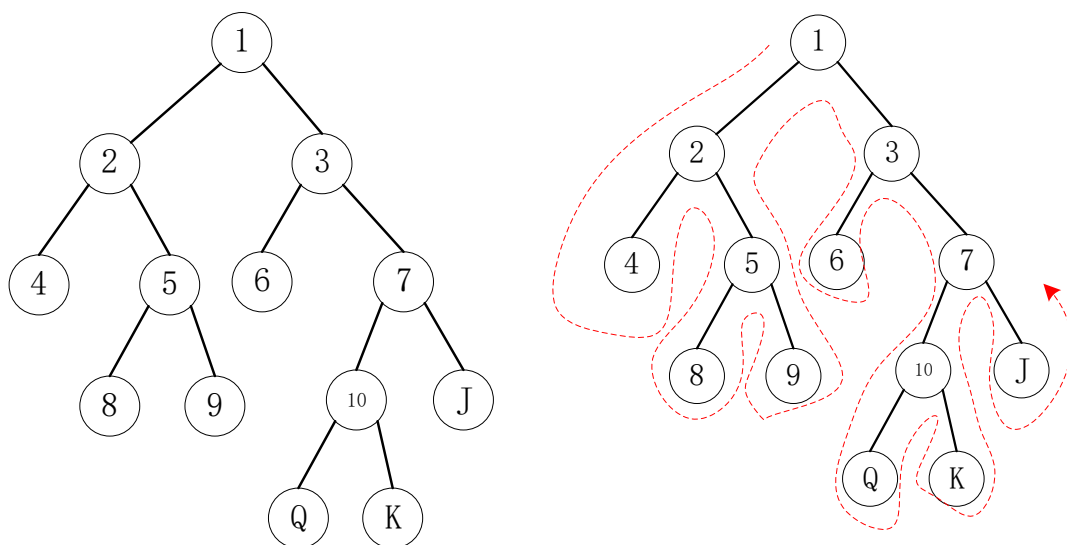
路径规划算法的常见衡量指标包括路径长度，时间，算法的时间复杂度以及代价等。其中，几个常见的指标解释如下：

- ①完备性：是指如果在起始点和目标点间有路径解存在，那么一定可以得到解，如果得不到解那么一定说明没有解存在；
- ②概率完备性：是指如果在起始点和目标点间有路径解存在，只要规划或搜索的时间足够长，就一定能确保找到一条路径解；
- ③最优性：是指规划得到的路径在某个评价指标上是最优的（评价指标一般为路径的长度）；
- ④渐进最优性：是指经过有限次规划迭代后得到的路径是接近最优的次优路径，且每次迭代后都与最优路径更加接近，是一个逐渐收敛的过程；

1.2.4 基于深度优先搜索(DFS)的三维路径规划方法

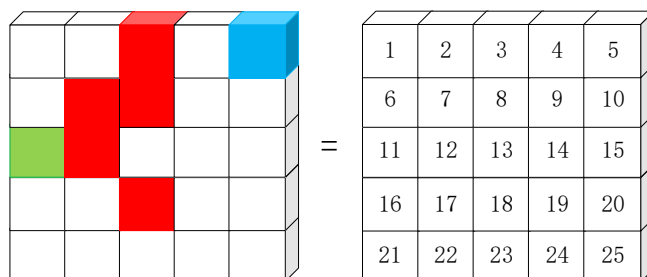
深度优先搜索算法(Deep First Search – DFS)作为一种遍历树、图的算法，可以应用于路径规划。DFS 算法可以对图进行尽可能深的搜索，利用递归函数嵌套的性质，可以将从搜索过程一直进行到终点。深度优先搜索的步骤如下

- ① 访问顶点 v
- ② 依次(规定左节点优先)从节点 v 的未被访问的邻接点出发，对图进行深度优先遍历；直至图中和 v 有路径相通的顶点都被访问
- ③ 若此时图中尚有顶点未被访问，则从一个未被访问的的顶点出发，重新进行深度优先遍历，直到图中所有顶点都被访问

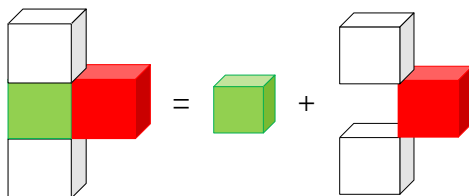


以下图为例，可以看出，搜索的路径为 1,2,4,5,8,9,3,6,7,10,Q,K,J

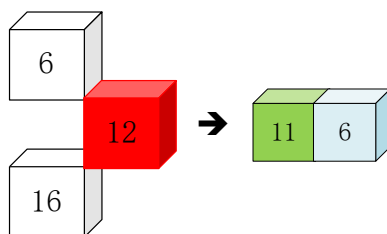
以简化的二维路径搜索图为例，假设起始点和终点的分布如下图所示



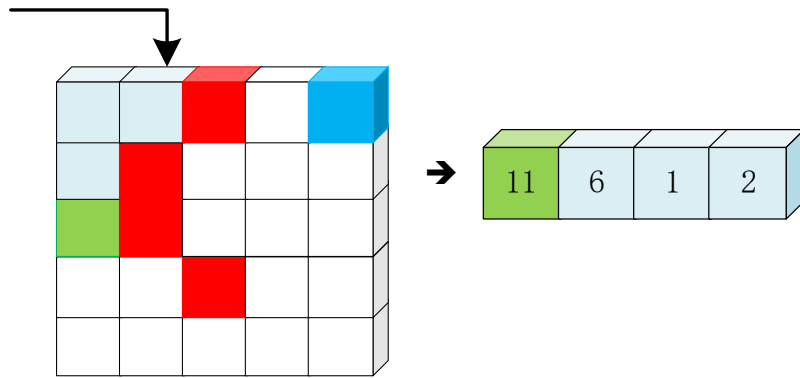
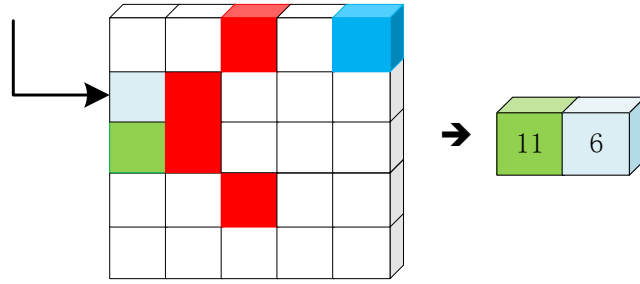
首先，根据起点的位置，分析其相邻节点的信息



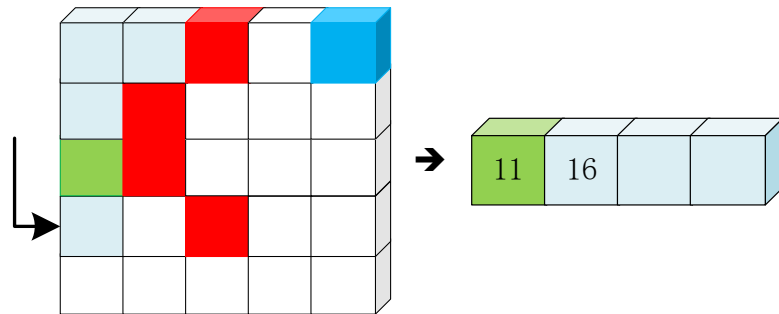
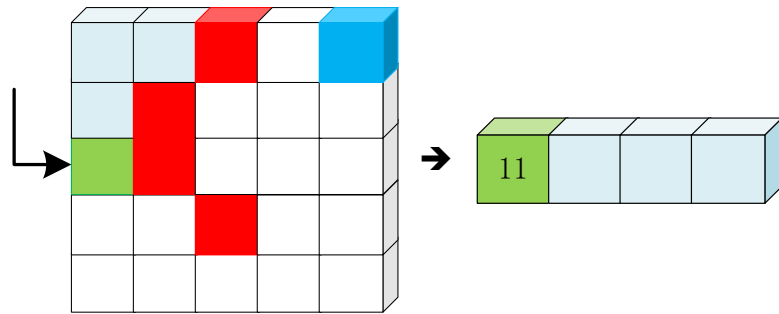
根据其相邻节点的信息，可以看出有两个方格为障碍物，故不能设置为下一步的位置。分析其他三个点与终点的位置关系，优先选择距离近的节点作为下一步



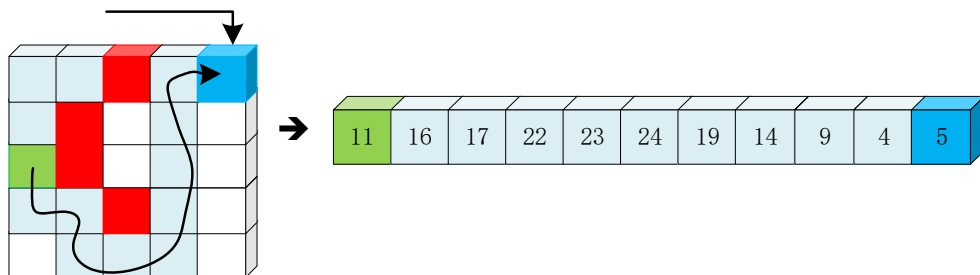
依次地，将步骤压入栈，并标记遍历过的点，使其区分于未遍历的点



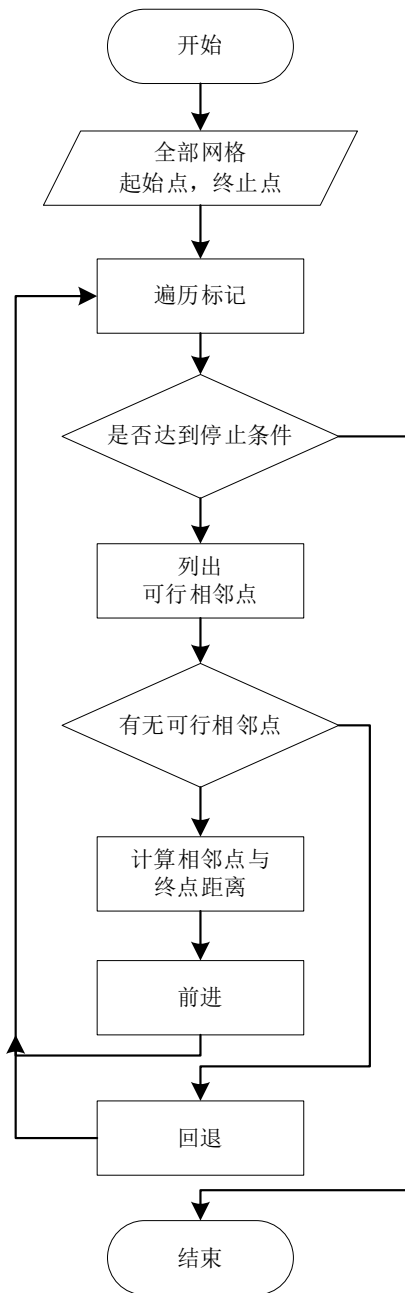
发现无路可走，则退回



.....



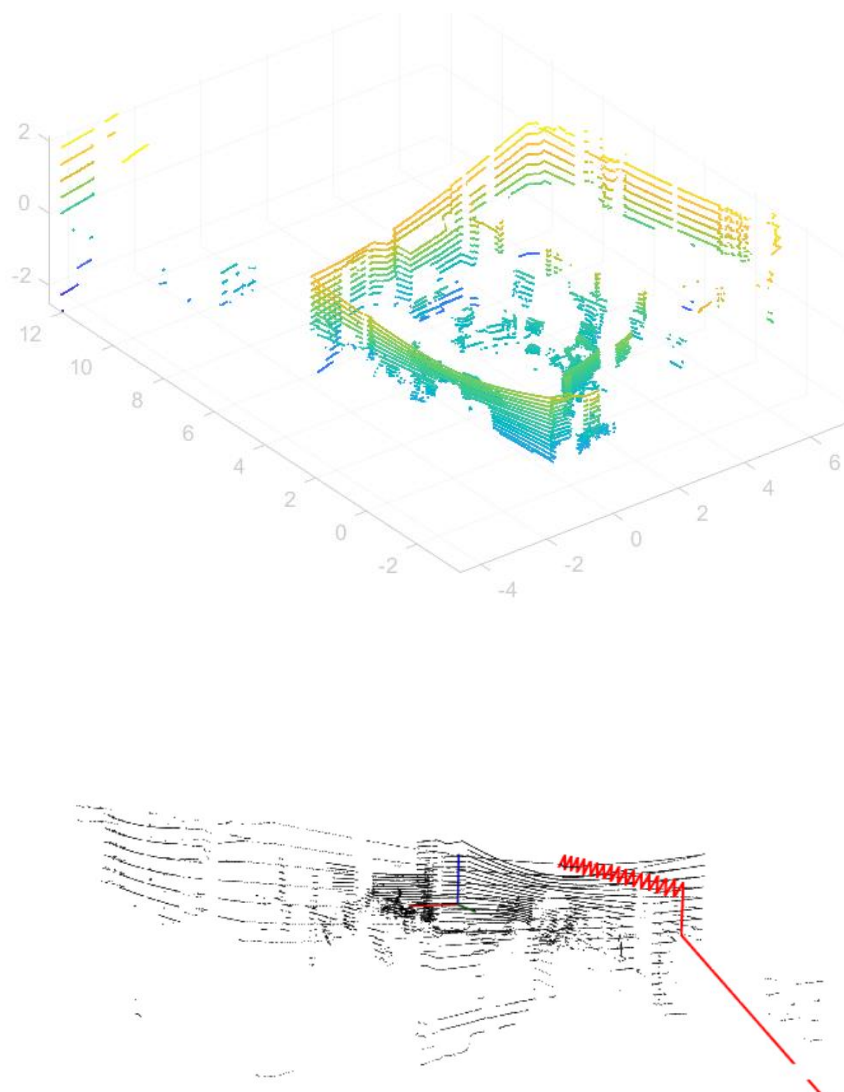
如上图所示为利用 DFS 算法进行路径搜索的过程。下面的流程图给出了 DFS 算法的步骤框图



```
void DFS(int * i)
输入: 起始节点地址 i, 停止条件, 路
径数组 route[]
int k=0
do
{
    if (route[k]-> neighborhood 不可行)
    {
        route.pop_back();
        k--;
    }
    else
    {
        route.push_back(节点);
        k++;
        DFS(k);
    }
}
while(~停止条件)
输出: 遍历路径
```

1.3 实验要求

根据某室内实录激光雷达点云数据，使用 PCL 库对点云进行初始化。并对数据进行必要的噪点、离群点去除等预处理，建立环境点信息。使用三角网格法或立方体法对环境点信息中的障碍物进行建模，并对立方体进行标号。使用给定的起始点和终点，利用 DFS 算法等完成路径规划并分析。得到路径后，分析路径的好坏并提出改进方案。建立 viewer 对象，对路径进行显示。



2 实验过程

1. 利用 PCL 库建立视窗对象 `viewer` 显示点云信息
2. 结合示例程序，在 C++ 工程中建立头文件，头文件名称为 `PathCalculate_学号后三位.h`，同时创建相应的 `cpp` 格式同名源文件，将 `PathCalculate` 封装成类
3. 根据原理中讲述的网格化过程对点云信息进行网格化建模，分析不同网格大小，网格形状对建模准确度的影响
4. 根据给定的起始点和终点，采用 DFS 算法计算路径
5. 输入点云数据，调试并运行程序。
6. 根据调试出现的错误，分析错误原因。
7. 若调试成功，请在 `viewer` 对象中显示规划的路径信息。

