张 文泰

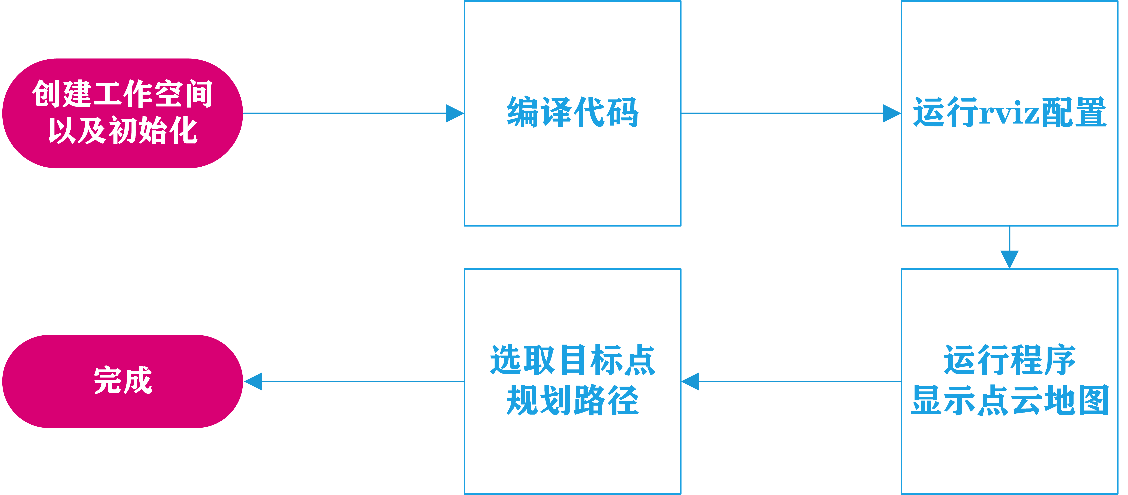
学号：21009101463 |

**机器人路径规划**

第二章作业

实验一 ROS版本作业

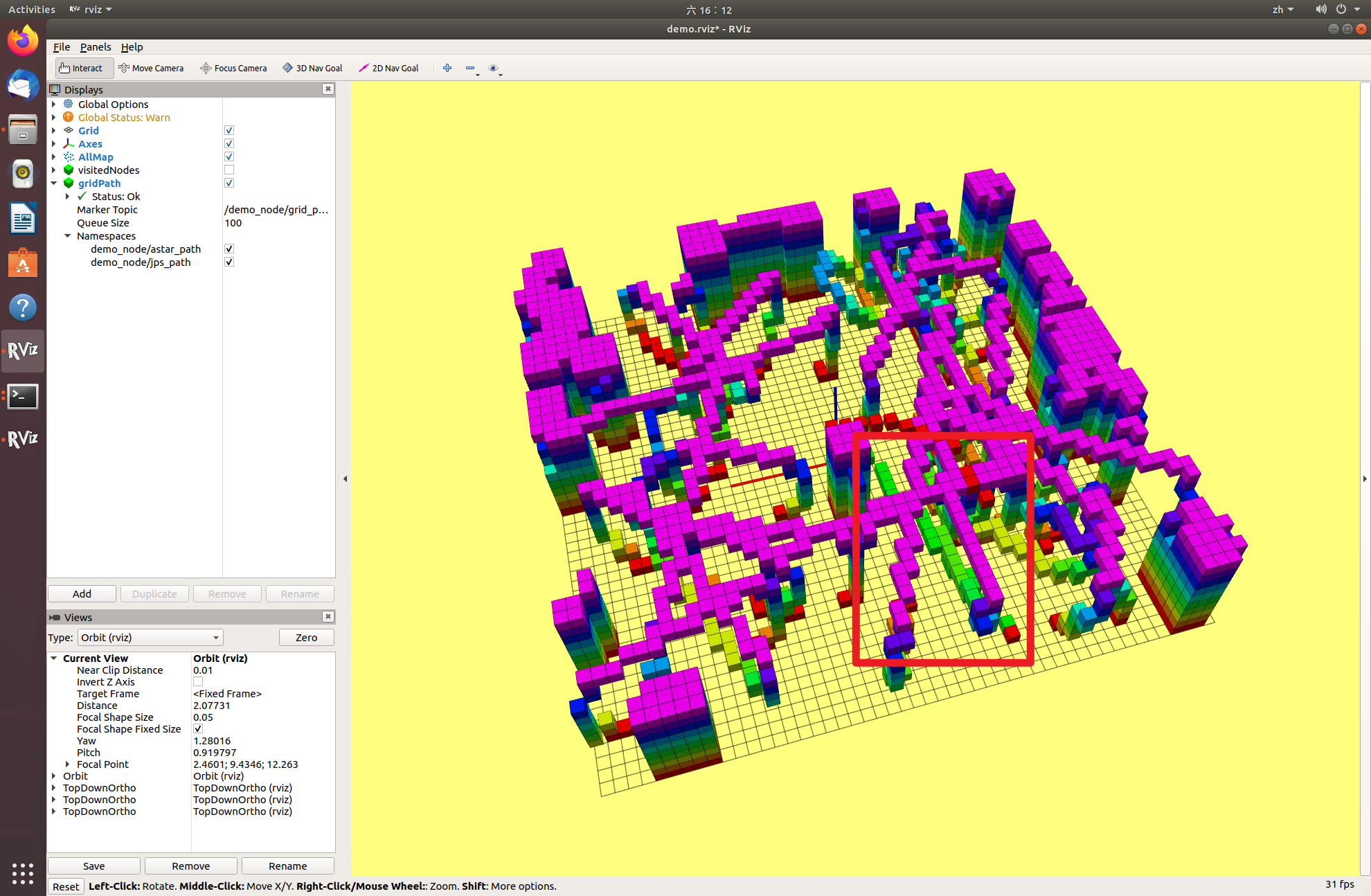
1. 实验过程



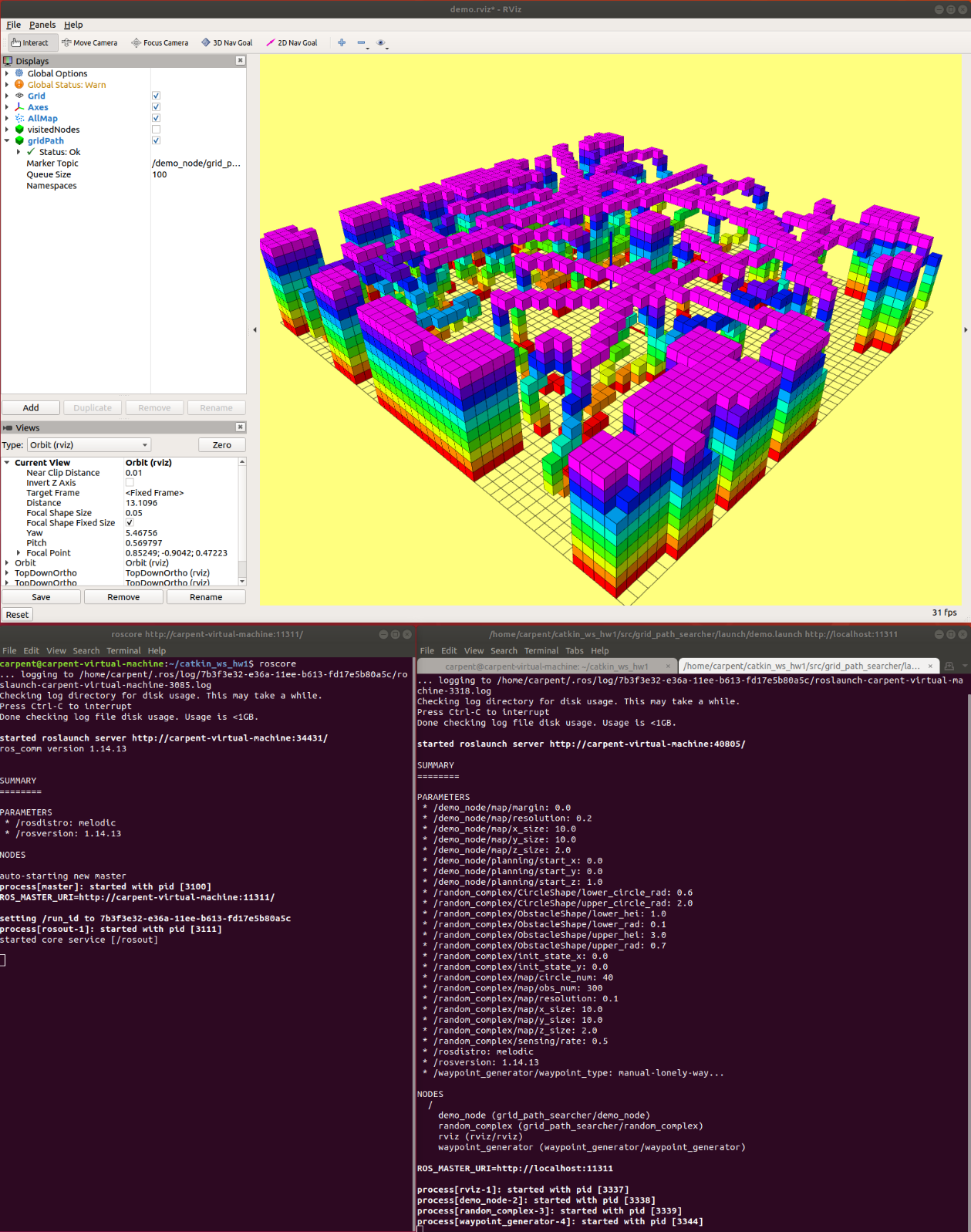
ROS版本作业操作流程图

1. 实验结果

路径规划结果如图所示：



A\*算法路径规划结果，红色框内为路径



地图运行结果及终端配置情况

实验二 MATLAB版本作业

1. 代码补充与分析

由于代码过长，因此分为两部分进行解析：



第一部分补充代码

整体来看，该部分代码通过一个迭代的搜索过程，不断更新当前节点和路径成本，直到找到从起点到终点的最短路径或者没有可行路径为止。下面是对该部分算法的逻辑框架梳理：

1. 初始化：循环开始前，xNode和yNode代表当前节点的坐标，xTarget和yTarget代表目标节点的坐标，NoPath作为是否找到路径的标志，为1表示还未找到，0则表示找到。

2. 循环条件：当前节点不是目标节点，并且NoPath = 1。

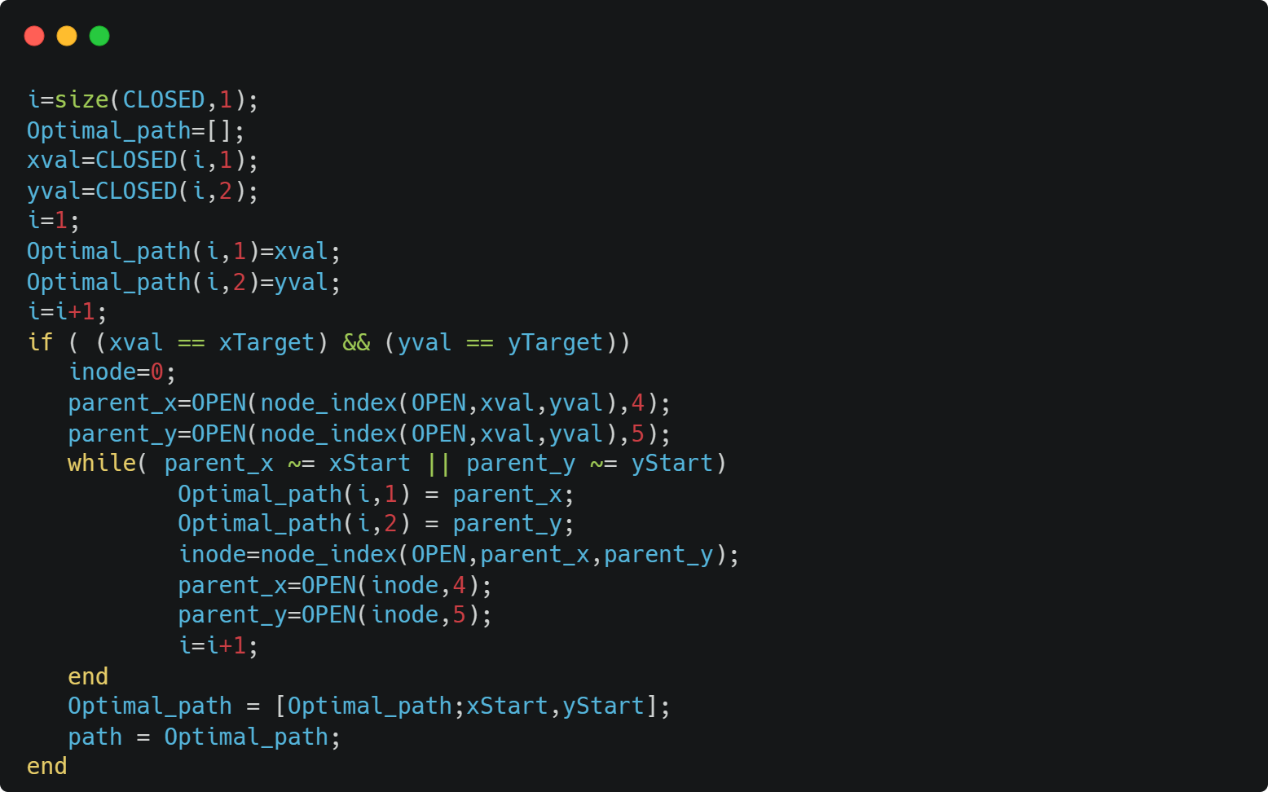
3. 扩展节点：使用expand\_array函数来获取当前节点所有可能的下一步移动，并将结果存储在exp\_array列表中。

4. 更新OPEN列表：OPEN列表用来存储被发现但还没被评估的节点，遍历exp\_array中的每个节点，检查它是否已经在OPEN列表中，如果在则更新该节点的成本和路径，如果不在则将其添加到OPEN列表。

5. 选择下一个节点：使用min\_fn函数从OPEN列表中选择成本最低的节点作为下一个节点。

6. 更新节点状态：如果找到了下一个节点，更新xNode和yNode为该节点的坐标，更新路径成本，并将该节点移动到CLOSED列表。如果没有找到下一个节点，设置NoPath为0，表示没有路径可找。

7. 结束循环：当前节点是目标节点或者NoPath为0时，循环结束。



第二部分补充代码

第二部分代码目的是通过路径回溯，从CLOSED列表中逐步构建出最优路径，下面是该部分代码的逻辑架构：

1.初始化变量：i设置为CLOSED列表的长度，即最后一个节点的索引。Optimal\_path数组用于存储最优路径。xval和yval设置为CLOSED列表最后一个节点的坐标，即目标节点。

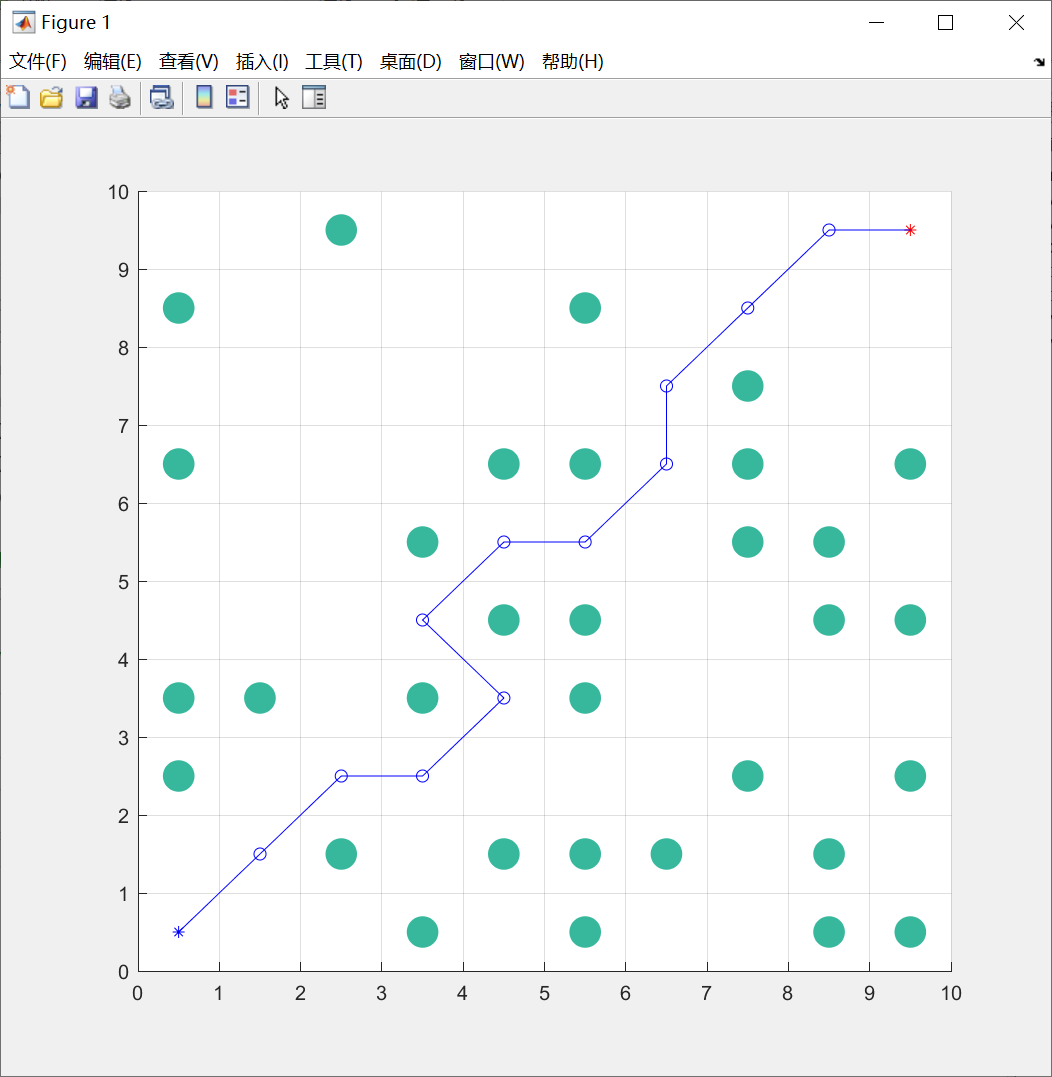
2. 构建最优路径：将目标节点的坐标添加到Optimal\_path数组的第一行。

3. 检查目标节点：如果当前节点坐标与目标节点坐标相同，则开始回溯过程。

4. 回溯父节点：使用node\_index函数在OPEN列表中找到当前节点的父节点坐标，随后将父节点坐标添加到Optimal\_path数组，并更新inode为父节点在OPEN列表中的索引。重复此过程直到父节点坐标为起始节点坐标。

5. 完成最优路径：将起始节点坐标添加到Optimal\_path数组的末尾，将最优路径存储在path变量中，用于在主程序visualize\_map函数进行绘图时调用。

1. 运行结果与感受



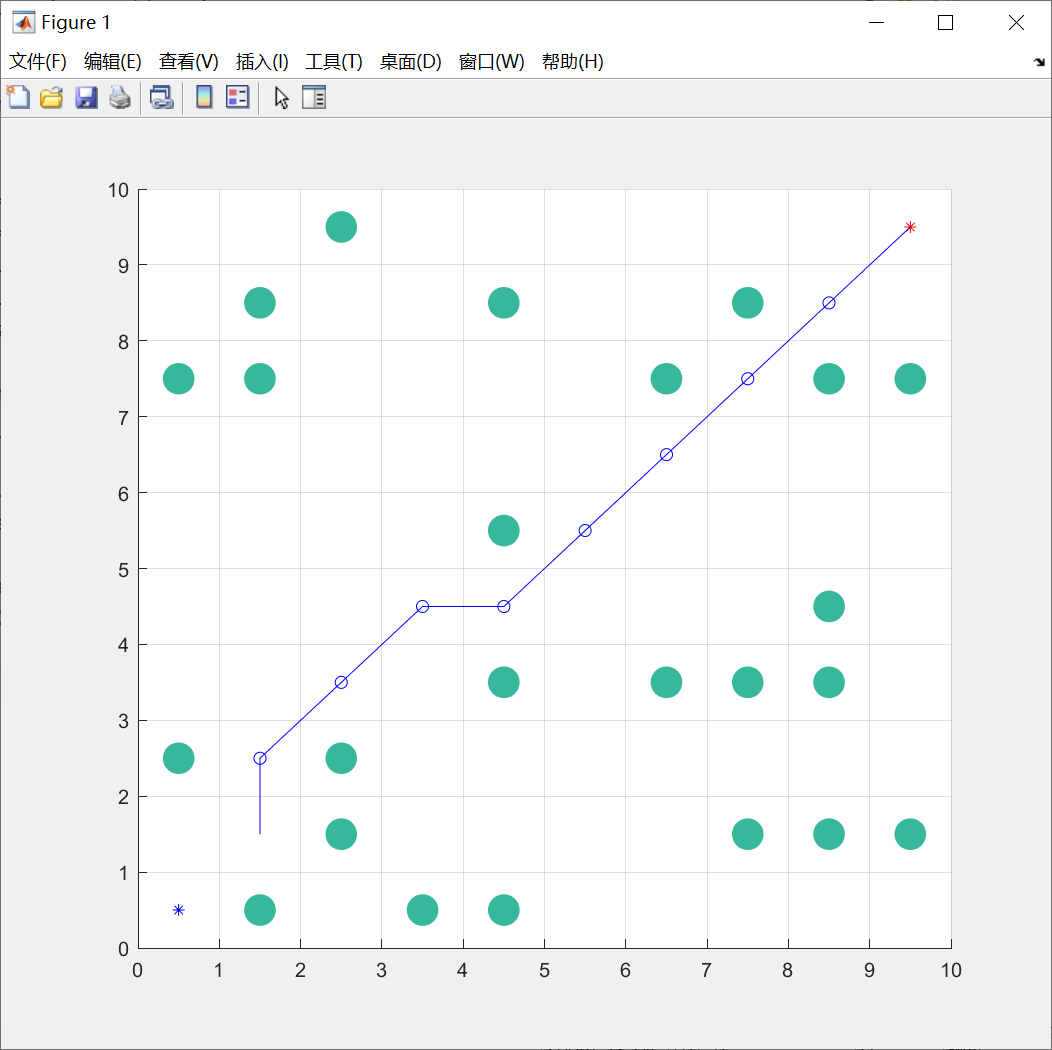
MATLAB版本作业运行结果

代码完成后运行多次，结果均正确，成功通过A\*算法实现了最优路径规划，本次实验也加深了我对A\*算法的认识与理解，锻炼了我的MATLAB代码能力与算法逻辑能力，让我对机器人路径规划课程知识有了新的感悟。

1. 问题与解决方案

**问题：**在初步完成作业代码补充之后，运行主程序，发现路径无法与起始点相连，如下图所示。

**解决方案：**在多次梳理代码结构，尝试改动代码之后发现，问题在于补充的算法代码在进行回溯时，最后没有将起始点包含在内，即起始点没有被添加到Optimal\_path数组中，因此在绘图时调用的path缺少了起始点坐标。随后在A\*算法最后添加了代码Optimal\_path = [Optimal\_path;xStart,yStart];成功解决了问题。



路径无法与起始点相连