**多移动机器人基于栅格地图的全覆盖探索技术报告**

# 问题描述

给定一个栅格地图，栅格地图中黑色为障碍物，白色为可通行区域。三个机器人从可通行区域出发，每个机器人在移动过程中可以对其自身周围一定范围内的区域进行探索，通过激光雷达等传感器得到该范围的环境信息。三个机器人需要基于栅格地图规划出一定的路径，从而对栅格地图所展示的范围实现全覆盖探索。

# 算法框架

机器人实现对栅格地图范围的全覆盖探索算法主要包含两大部分，一部分是地图的预处理，另外一部分是机器人路径规划。

首先是对地图进行预处理。机器人自身有一定的宽度，因此在构建覆盖区域时需考虑机器人的尺寸，本文主要采取的是将地图中的障碍物膨胀一个机器人宽度以消除机器人尺寸带来的影响。在进行路径探索之前，由于有一些可通过区域是被障碍包围着的，机器人无法到达，因此可以将其直接标注为黑色障碍物，在搜索的时候可以大大减少工作量，本文采取的方法是基于广度优先搜索算法的不可到达区域标注。机器人利用栅格地图覆盖某一区域时，需要考虑地图分辨率对其影响。栅格地图分辨率指一个栅格在实际工作环境中所表示的距离，距离越小，分辨率越高，即实际环境信息描述越详细，精度越高，但分辨率过高导致探索效率低下；而分辨率越低，即直接使用低精度栅格地图来进行路径规划，造成障碍物信息描述不完全，造成覆盖率下降。而原栅格地图的分辨率过高，因此需要对栅格地图进行一定的缩小，但是缩小不宜过多。

地图预处理算法框架如图2. 1：

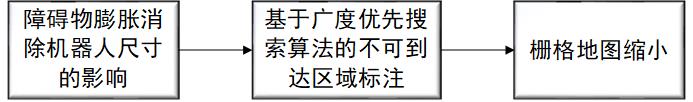


图2. 1 地图预处理算法框架

另外一部分是机器人路径规划。机器人需要根据已经探索的区域和未探索的区域进行目标点的寻找和选取，然后规划出一条到目标点的最优路径，直到找不到目标点为止，本文采取的是基于到达成本的目标点选取以及基于A\*算法的路径规划。

机器人路径规划算法框架如图2. 2：

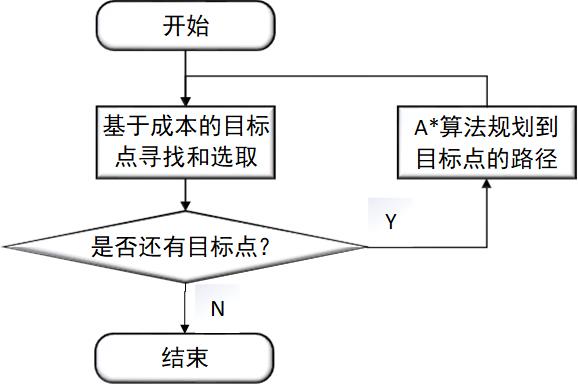


图2. 2 机器人路径规划算法框架

具体算法实现将在第3部分详细介绍。

# 算法实现

## 地图预处理算法

### 障碍物膨胀

图像膨胀是图像处理中的重要方法之一，其作用是将图像中原本断裂开来的同一物体桥接起来，同时可以在一定程序上扩大物体的尺寸。在本研究中，由于输入的二维栅格地图障碍点比较离散，经过一个机器人宽度的膨胀之后，离散的障碍点形成了一个整体，同时，对于机器人来说，白色可通行区域变得更窄，消除了机器人本身尺寸带来的影响，如图3. 1 a),b)所示：



a)膨胀前图像 b)膨胀后图像

图3. 1 膨胀前后图像的变化

### 基于广度优先搜索算法的不可到达区域标注

广度优先搜索算法（Breadth-First Search，简称BFS）是一种用于遍历或搜索树或图的算法。它按照树的层次，或者图的层级，逐层访问节点。这种算法首先访问起始节点，然后遍历起始节点的所有邻居节点，接着再遍历这些邻居节点的未被访问过的邻居节点，如此逐层扩展，直到所有可达的节点都被访问过。

广度优先算法的基本步骤如下：

（1）初始化：

创建一个队列Q，并将起始节点s放入队列中。同时，创建一个集合visited，用于记录已经访问过的节点，初始时将起始节点s加入visited。

（2）循环处理：

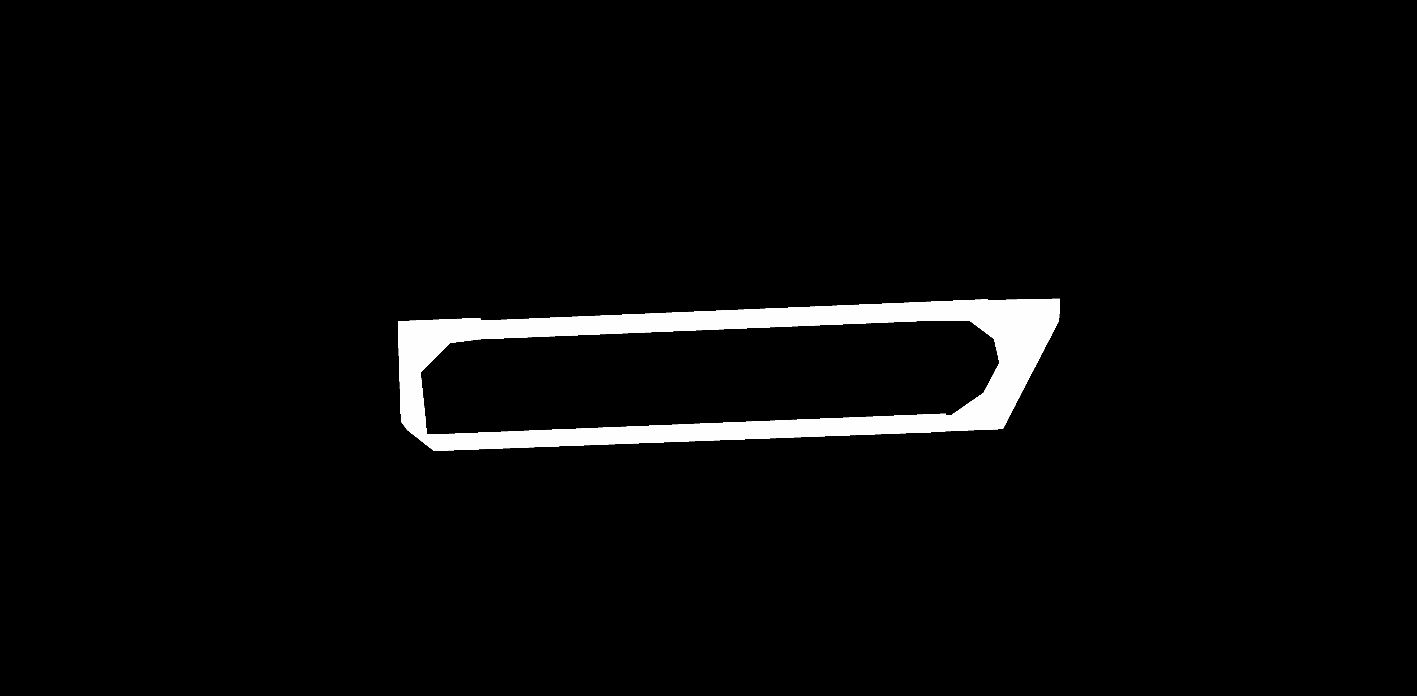
当队列Q不为空时，执行以下步骤：

a.从队列Q中取出一个节点n。

b.遍历节点n的所有邻居节点m。对于每一个未被访问过的邻居节点m，执行以下操作：将节点m加入队列Q，并将节点m加入visited集合。

c.结束：当队列Q为空时，表示所有可达的节点都已经被访问过，算法结束。

基于栅格地图，使用广度优先算法，以机器人初始所在的点开始向外扩展，把所有可到达区域都标注为白色可通过区域，所有不可到达区域都标注为黑色障碍物，可以有效地加快后续探索效率。将机器人初始位置定在中间的圈内，不可到达区域标注前后图像如图3. 2 a),b)所示：

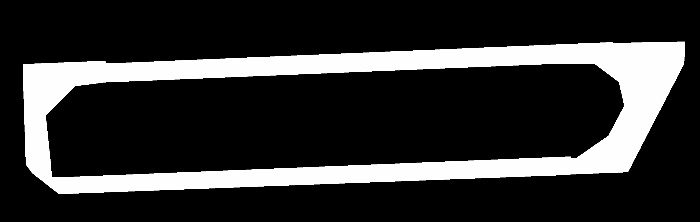


a)标注前图像 b)标注后图像

图3. 2 不可到达区域标注前后图像变化

### 栅格地图缩小

在本研究中，原栅格地图的像素为(5705,7028),在该栅格地图下进行探索路径规划，尽管规划的路径将会更加精确，但是耗费的算力资源极大。实际上，将地图像素缩小10倍之后，原地图的障碍信息并没有变化太多，以该地图作为输入进行后续的机器人探索，所使用的时间将会减少10的平方，也就是100倍。但是，地图是不能随意缩小的，当地图缩小到一个机器人的尺寸为1个像素点（缩放50倍）的状态时，其障碍信息及可通行区域的信息将会变得模糊，不适合后续的探索。未缩小之前，缩小10倍，缩小50倍的某一区域图像效果如图3. 3 a),b),c)所示。因此，本研究选择对原栅格地图缩放10倍，同时为了保证缩小前后依然是纯黑色为障碍，纯白色为可通过区域，采用最近邻插值算法来进行图像缩放。



a)未缩小之前 b)缩小10倍 c)缩小50倍

图3. 3 栅格地图缩小前后变化

## 机器人路径规划算法

### 基于到达成本的目标点选取

为了使机器人能够对地图进行全覆盖探索，首先需要初始化一张地图，该地图与原栅格地图的尺寸相同，但全部为未探索区域，未探索区域的像素值为205。机器人从某一个位置出发，每一步都会对其周围的一定区域进行探索，已被探索的区域将会变为原栅格地图对应的像素值，同时以自身的位置为参考通过广度优先搜索（BFS）构建一个与原栅格地图尺寸相同的距离地图，表示到图中所有白色可到达区域所要走的曼哈顿距离。每一个机器人都会根据已探索区域和未探索区域的分界选取目标点集，将所有目标点都存储在一个列表中。接下来将会选取该列表中到达成本最小的一个点作为最终的目标点，目标点到达成本的定义如下：



其中表示到达该目标点的曼哈顿距离，是在到达自身目标点的过程中，所碰到的其它机器人所经过的路径点个数，设置这一项成本是为了让在有多个机器人的时候，使它们有一定的任务分配，减少重复向同一个方向去探索，提高探索效率。

### 基于A\*算法的路径规划

选取了目标点之后，需要根据地图来规划出一条从起始点到目标点的路径，本研究采取的是A\*算法的路径规划。A\*算法是一种广泛应用于路径搜索和图搜索的启发式算法。它结合了广度优先搜索和贪婪最佳优先搜索的优点，能够高效地找到从起点到目标节点的最短路径。

A\*算法的基本原理如下：算法中引入估价函数，其中，是起始结点到结点n目前所知的最短路径值，是结点n到目标的估价启发值。算法中用到了两个链表,一个是开放列表openlist, 用来存储待检查的栅格单元数据；另一个是关闭列表closelist，用来保存所有不需要再次检查的栅格单元。算法过程描述如下：

（1）创建开放列表openlist, 初始化使其只包含起始点;

（2）创建关闭列表closelist，初始化清空;

（3）从openlist中选择值最小的一个节点n;

（4）如果n是目标终点，停止搜索;

（5）对于n的4个相邻节点中的一个节点：

a.如果m在closelist中并且更小, 更新节点m的值, 将其父节点指向n

b.如果m在openlist中并且目前的更小，更新节点m的值, 将其父节点指向n

c.如果m不在openlist和closelist中，将m加入openlist中，计算其值，将其父指针指向n

（6）返回执行（3），继续搜索;

（7）从终点向上回溯到起点, 记录经过的栅格节点。

# 实验结果及分析

运行设备：华硕天选4 i9-13900H

运行环境：Linux + python3.11

输入：一张二维栅格地图，其尺寸为（5705，7028），三个机器人初始所在位置坐标，如图4. 1所示（因原图尺寸过大,该图仅展示的是进行路径规划的部分区域）：



图4. 1 输入二维栅格地图及机器人初始所在位置

输出：三个机器人的路径，以及覆盖探索效果图，如图4. 2所示：

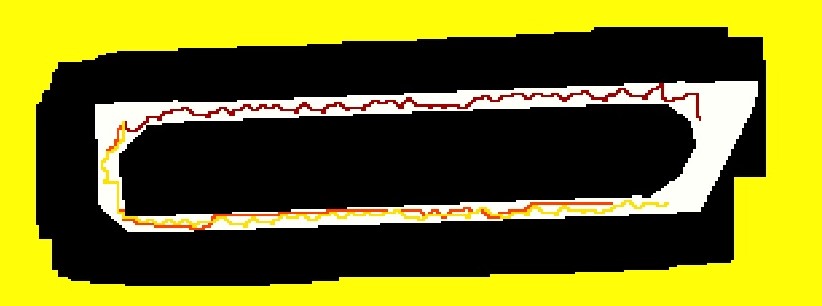


图4. 2 机器人路径和覆盖探索效果图

运行时间：图像预处理总用时：1分50秒 路径规划用时：2分12秒

# 总结

本研究实现了输入一张二维栅格地图和多机器人初始所在坐标，输出多机器人完成对栅格地图所展示的区域的覆盖探索所经过的路径点坐标，为后续SLAM建图任务提供了路径点参考。