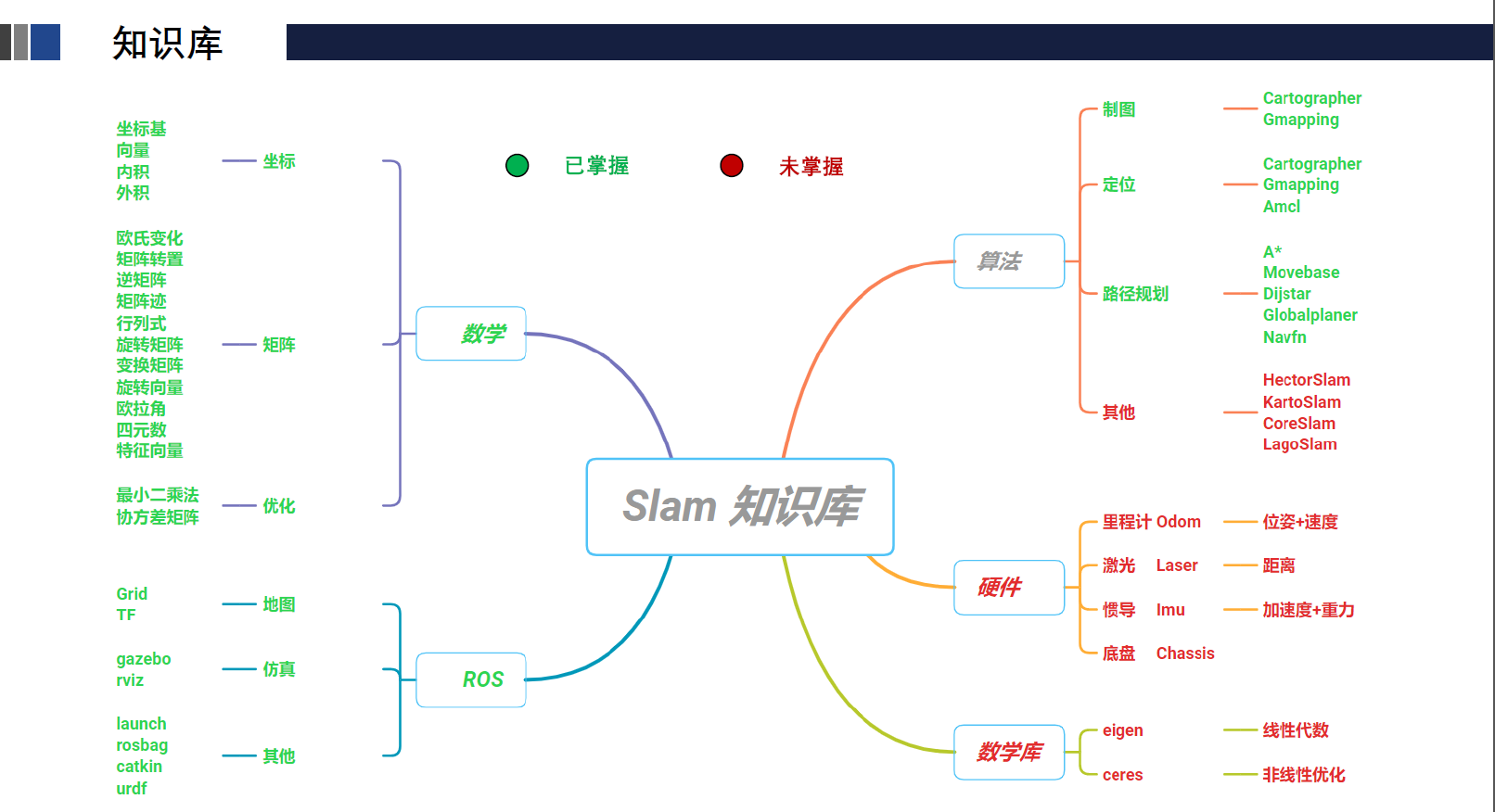
SLAM知识库

2021/12/27：Slam知识库



1. 目标：完成栅格地图的优化：



2021/12/28：搭建 Linux 开发环境"VMwareWorkstation 16 player"+"Ubuntu 18.04"；

搭建 ROS(melodic) 开发环境，包含 rviz，gazebo 等工具；  
2021/12/29：熟悉 ROS 栅格地图格式，占据栅格地图消息结构 OccupancyGrid；  
优化栅格地图，对 Slam 建图有未知区域，边缘不清晰栅格进行优化；

ROS官网：[rviz](http://wiki.ros.org/rviz)/[DisplayTypes](http://wiki.ros.org/rviz/DisplayTypes)/[Map](http://wiki.ros.org/action/fullsearch/rviz/DisplayTypes/Map?action=fullsearch&context=180&value=linkto:"rviz/DisplayTypes/Map"" \o "点击对这个标题进行全文检索):http://wiki.ros.org/rviz/DisplayTypes/Map

ROS发布栅格地图消息类型：nav\_msgs/OccupancyGrid

使用命令rosmsg show nav\_msgs/OccupancyGrid查看消息的类型

std\_msgs/Header header

uint32 seq

time stamp

string frame\_id

nav\_msgs/MapMetaData info

time map\_load\_time

float32 resolution //栅格地图分辨率对应栅格地图中一小格的长和宽

uint32 width

uint32 height

geometry\_msgs/Pose origin

geometry\_msgs/Point position

float64 x

float64 y

float64 z

geometry\_msgs/Quaternion orientation

float64 x

float64 y

float64 z

float64 w

int8[] data

它主要由map\_msg.header, map\_msg.info, map\_msg.data 三个主要信息，map\_msg.header包含了发布地图的序列号（seq），时间戳（stamp）和帧id信息；map\_msg.info包含了发布的地图的加载时间，分辨率，宽度，高度，地图起点坐标和方向；map\_msg.data 数组包含了地图的元数据信息，可以看到，它是int8类型的，也就是说每个元数据的值范围是-128~127之间，在链接http://wiki.ros.org/map\_server 中提到，

map\_msg.data 数组的 值100表示单元被占用；值0表示空闲，即未占用；-1（无符号类型下的255）表示未知。我们知道，这里不同的值含义不同，它们在rviz可视化界面上显示的颜色也不同，如下图所示

int a[4]={0,1,2,3,3}; vector<int> v5(a,a+5);//v5的size为5，v5被初始化为a的5个值。后一个指针要指向将被拷贝的末元素的下一位置。

**导航：**ROS通过OccupancyGrid(占据网格)进行导航，OccupancyGrid由一个xxx.yaml格式的元数据文件，和图片格式的地图数据文件组成。

地图元数据 xxx.yaml ：

image: testmap.pgm

resolution: 0.1

origin: [0.0, 0.0, 0.0]

occupied\_thresh: 0.65

free\_thresh: 0.196

negate: 0

image: 指定包含occupancy data的image文件路径; 可以是绝对路径，也可以是相对于YAML文件的对象路径 。

resolution: 地图分辨率，单位是meters/pixel 。

origin: 图中左下角像素的二维位姿，如（x，y，yaw），yaw逆时针旋转(yaw=0表示没有旋转)。系统的很多部分现在忽略yaw值。

occupied\_thresh: 像素占用率大于这个阈值则认为完全占用。

free\_thresh: 像素占用率比该阈值小被则认为完全自由。

negate: 无论白色或黑色，占用或自由，语义应该是颠倒的（Whether the white/black free/occupied semantics should be reversed (interpretation of thresholds is unaffected))

3.利用map\_server将地图保存在本地

rosrun map\_server map\_saver [--occ <threshold\_occupied>] [--free <threshold\_free>] [-f <mapname>]

地图数据：

Image 描述了地图上每个单元在相应像素的颜色中的占用状态。白色像素表示自由，黑色像素格表示占用，两种颜色之间的单元表示未知。彩色和灰度图像都可以，如果是彩色图像，则计算所有通道的平均值。图像单元占用概率的计算如下：occ = (255-color\_avg)/255.0 （color\_avg是所有通道的平均值），该公式表示像素点越黑占用概率越高，越白占用概率越低。

当ROS消息通信时，数据类型为nav\_msgs::OccupancyGrid，此时占用度被表示为范围为[0-100]的整数，0的意思完全是自由的，100的意思完全占用，特殊值-1完全未知。

原文链接：https://blog.csdn.net/qq\_37394634/article/details/119818976

C++中变量的定义与声明

变量的定义（definition）用于为变量分配存储空间，还可以为变量指定初始值。在程序中，变量有且仅有一个定义。

声明（declaration）用于向程序表明变量的类型和名字。定义也是声明：当定义变量的时候我们声明了它的类型和名字。可以通过使用extern声明变量名而不定义它。不定义变量的声明包括对象名、对象类型和对象类型前的关键字extern。

extern是声明不是定义，也它不分配存储空间。事实上它只是说明变量定义在程序的其他地方。程序中变量可以声明多次，但只能定义一次。

C++判断：

1. 知道具体的循环次数使用for循环；
2. 只进行单次判断使用if判断；
3. 如果进行多次判断可使用While循环进行判断；

2021/12/29：ROS学习（1 ~ 7）讲

动态参数：https://blog.csdn.net/u011832219/article/details/109447075

ROS的另一个功能是动态参数配置应用，如果使用PID节点调试电机时，就应该使用动态参数配置应用；

2021/12/30：ROS学习（8 ~ 12）讲

xacro文件：https://blog.csdn.net/chishuideyu/article/details/53

Xacro (XML Macro)是一种XML宏语言。何使用xacro来简化urdf文件。

2021/12/31：栅格地图解析

OccupancyGrid栅格地图每个格子使用占用率表示，颜色从白到黑，范围【0,100】的概率表示占用率，-1表示未知区域。描述2D栅格地图，

消息定义格式如下：

Std\_msgs/Header header 数据头

Nav\_msgs/MapMetaData info 地图参数

Int8[] data 地图的数据

\* 向Rviz发送的数据中一定要包含map.header.frame\_id = “map”;

栅格地图的初始大小：

map.info.width = 20; map.info.height = 20； （宽和高）

栅格地图分辨率对于栅格地图中一格的长和宽：

Map.info.resolution = 0.5;

问题：

1. 在机器人建图（gmapping算法）的同时优化地图，还是建图后得到pgm地图文件后优化地图？

保存pgm图片之后对图片进行处理；

2022/1/4：pgm图片的数据格式

PGM图片是便捷式灰度图像（protable gray map），图像格式分为两类：P2和P5类型。

PGM图片的数据格式为：头文件 + 数据

头文件：PGM文件格式（P2和P5）；图像的宽度；图像的高度；图像灰度值可能的最大值；

数据：记录图像每个像素的灰度值；顺序为从上到下，从左到右描述；且P2格式与P5格式不同；

P5格式的文件：每个像素用二进制表示。

P2格式的文件：每个像素用字符串表示。

学习VS Code的使用方法。通过VS Code读取PGM图片的地图数据；

从数据的角度出发，分析和优化地图数据；

1. 安装VS Code软件和相关插件，Remote-SSH配置，samba服务器配置。
2. 安装MinGW软件和相关插件。
3. 分享会：lululemon发展的故事；结论：完美主义，做到极致，不将就；
4. 深入客户的场景，做到与客户感同身受，让我产生的想是：心理咨询师

2022/1/5：ROS入门21讲（古月）

使用C++程序读取pgm图片的数据，但是数据乱码，因为pgm图片的数据是二进制连续的数据，图片robot\_map.pgm的图片数据：

P5

9 10

255

������������������������������������������������������������

在软件下打开时，数据部分的值：

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

FE FE FE FE E9 18 18 18 FF

2022/1/6：使用二进制的方式读取pgm图片的数据 /ROS入门21讲（古月）

查找处理方法：

方法1：把前面的文件头分离出来，只对二进制数据部分进行处理；（没法处理）

方法2：直接以二进制的方式读取文件数据然后一起处理；（可行）

方法3：使用数字图像处理（边缘检查、噪点清除、检测直线，高斯滤波）

2022/1/7：总结：完成的部分

1、熟悉开发环境，了解机器人导航功能，对系统有初步的了解

- 搭建 Linux 开发环境 "VMware Workstation 16 player" + "Ubuntu 1804"

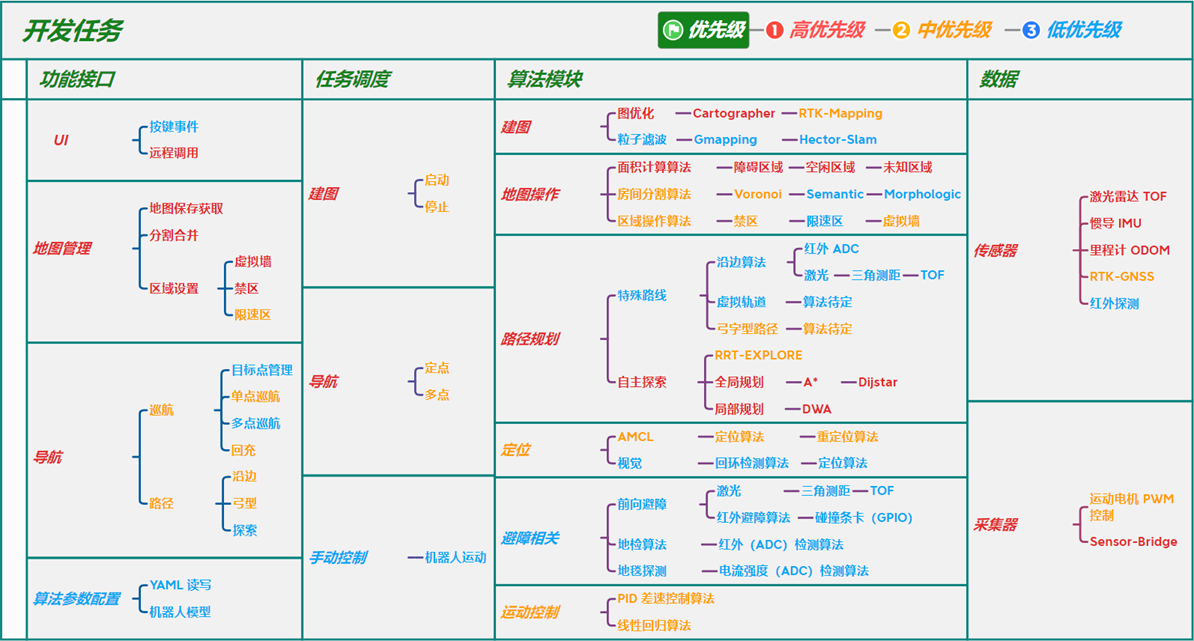
- 搭建起 ROS(melodic) 开发环境，包含 rviz，gazebo 等工具

- 熟悉 ROS 栅格地图格式，占据栅格地图消息结构 OccupancyGrid

- 学习ROS的基础知识，ROS入门21讲和ROS学习博客，rviz导航建图；

- 安装samba、visual studio code、remote-ssh；

浪费了很多时间：没有责备，没有为难；



2022/1/10：方法2：直接以二进制的方式读取文件数据然后一起处理；

步骤1：读取robot\_map.pgm的数据，以十六进制的方式显示处理；

步骤2：以列表的形式将数据保存到data\_list列表中，

2022/1/11：方法2：用C语言读出图片数据并进行修改；效果不理想；

像素值：与阈值设置

image[0][0] = 205 灰色

image[3][8] = 184

image[4][8] = 179

image[10][8] = 199

image[15][8] = 200

黑色 <150

image[1][18] = 133

image[3][18] = 128

白色 > 220

image[2][18] = 254

image[15][17] = 220

1. 从OccupancyGrid栅格地图数据：

Int8[] data 地图数据出发进行优化；

1. 调研查找方案：地图划分，面积计算；

2022/1/12：房间分割的程序：ipa\_coverage\_planning/ipa\_room\_segmentation

1. 对于大型地图，算法可能需要几分钟才能完成分割，尤其是基于特征的和 voronoi 随机场分割。如果你想要快速的结果，形态和距离分割是最快的，但如果你有广阔的空间，它们可能不会返回最好的结果。
2. Voronoi 随机场分割：首先为给定的地图计算一个修剪过的 voronoi 图，在该图上，一个邻域被集中到一个中心点。找到所有点后，就获得了相邻关系，这意味着哪个节点与另一个节点相邻，具体取决于 voronoi 图。这会产生几个派系，从而产生一个条件随机场。这个 CRF 的特征是AdaBoost分类器对房间、门口和走廊的弱响应。这些AdaBoost分类器使用与基于特征的分割相同的特征。然后在这个 CRF 上进行循环信念传播，产生几个相交的片段。
3. 面积计算：利用房间分割的方法，获取每个房间的大小，然后计算每个房间占多少个像素，根据像素求解房间的面积；

2022/1/13：分析建图的过程

建图之后发布栅格地图的数据，如果只对数据部分进行处理，（很难）

\*仿真扫描地图，预先给定的是4000 \* 4000的个像素，地图尺寸为200 \* 200（m）

分辨率为：0.05（m/pixel）

1. 先设置地图大小m \* m米和分辨率：0.05（m/pixel），设置机器人开始的建图位置；
2. 然后计算地图像素大小p \* p像素，
3. 扫描完地图后，data的数据就是p \* p个
4. 保存为地图后对图片进行优化；（使用opencv图像处理）
5. 使用Voronoi 随机场分割对地图进行房间分割，
6. 利用每个房间占据的像素个数计算房间的面积；（opencv计算轮廓面积）
7. pgm图片转png图片；

2022/1/14：ROS学习（1-6）章

1. C++学习：::被称为作用域符号(scope resolution operator)．

命名空间： 在我们之前的ros程序中，遇到了两个命名空间，一个是std\_msgs，另一个是geometry\_msgs．Int8, Float64等都是std\_msgs这个命名空间下的类，PoseStamped等是geometry\_msgs这个命名空间下的类． 回到上面的程序我们在定义完phd这个命名空间后，可以使用using namespace phd，这样在main函数中我们可以不使用phd::来定义一个phd下的student类的对象,直接student phdStudent即可．同样，如果我们添加using namespace master，我们也可以直接使用student masterStudent来定义msater命名空间下student类的对象． 但是如果在程序中同时添加了

using namespace phd;

using namespace master;

这时候你在main函数中写student object\_name就肯定会报错．因为电脑无法知道你要使用的student类是属于哪个命名空间的．所以一般为了图方便，在我们确定没有类名会重复时，我们添加using namespace ...这一行在定义完头文件之后，这样我们就可以省去在定义类时一直使用namespace\_name::类名这种格式命名．但是有些时候如果两个库很有可能有相同的类名，就不要使用using namespace ...，不然很有可能造成程序的误解．

模板：模版是为了避免重复定义同样功能的函数而开发的．

1、普通的类的对象的定义格式如下类名　 对象名(构造函数参数)

2、模版下的类的对象定义的格式就是 类名<模版变量类型> 对象名(构造函数参数)

3、 当我们遇到参数很多的程序时，我们需要使用yaml文件。yaml文件是使用一种特定结构来储存外部参数的文件。

2022/1/17：ROS学习（6-10）章

https://www.guyuehome.com/author/%E9%99%88%E7%93%9C%E7%93%9C

1. 主函数中，ros::NodeHandle private\_node("~")和我们平时定义node的方式不是很一样。平时我们定义node都是直接ros::NodeHandle nh。这里多了一个参数～。它表示此时这个nodehandle是读取局部参数的

#开始学习c++图像处理：

网址：https://blog.csdn.net/weixin\_44753371/article/details/114786481  
环境 linux + ubuntu18，OpenCV4.4.5；

1. 图像的读入：imread()函数，即为图像的读入；
2. 图像的修改：cvtColor()函数，通常需要将BGR格式转换为其他格式，如HSV、HLS等。
3. 图像的保存：imwrite()函数，将修改或处理后的图像保存在自己的文件夹中。

2022/1/18~ /19：c++图像处理

1. c++取图像的像素：img.at操作。
2. 对图片进行处理：实现功能：模糊、膨胀和腐蚀；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 图片 | 原图 | 阈值+腐蚀 | 阈值+腐蚀+中值 | 阈值+腐蚀+中值+膨胀 |
| map1.pgm | 原图 | 阈值+腐蚀处理 | 阈值+腐蚀+中值处理 | 阈值+腐蚀+中值+膨胀处理 |
| map2.pgm | 原图 | 阈值+腐蚀处理 | 阈值+腐蚀+中值处理 | 阈值+腐蚀+中值+膨胀处理 |
| Demo1.pgm | 原图 | 阈值+腐蚀处理 | 阈值+腐蚀+中值处理 | 阈值+腐蚀+中值+膨胀处理 |

\*是否存在处理误差；精度问题；

2022/1/20：上午：c++图像处理 下午：新员工培训

基础知识：Mat操作（创建图像数组）；Mat - The Basic Image Container

1、像素img.at(row,col)[1]获取和修改；

2、图片中绘制直线、圆、矩形等，利用阈值把绘制的图形转换为白色，利用膨胀和腐蚀进行修复；

3、均值模糊：主要目的是为了降低图像中的噪声，

中值滤波：中值滤波对椒盐噪声有很好的抑制作用高斯模糊：

高斯双边模糊：高斯双边模糊可以避免边缘信息的丢失，保留图像的轮廓

4、膨胀：以最大值替换重叠图像下锚点的像素，原图白色较多(255,255,255), 黑色(0,0,0)区域较少,根据膨胀原理，将结构元B对应A的区域中最大的值赋值给锚点，即将(255,255,255)像素值赋值给锚点，即膨胀后白色的区域增加，黑色的区域减少。

5、腐蚀：以最小值替换重叠图像下锚点的像素。  
腐蚀前白色(255,255,255)区域偏多，进行腐蚀操作，即将最小值赋值给锚点像素，最小值黑色(0,0,0)赋值给锚点，即黑色区域增大。

2022/1/21：c++图像处理

1. 开运算：先腐蚀后膨胀：先进行腐蚀，腐蚀就是将最小值像素赋值给锚点，即将黑色像素值赋值给锚点，除去了白色点，后进行膨胀，将最大值即白色点赋值给锚点。
2. 闭运算：首先进行膨胀操作，即将最大值白色像素赋值给锚点，可以除掉较大的黑色像素值，后进行腐蚀操作，即最小值黑色像素赋值给锚点。
3. 顶帽：顶帽 = 原图像 - 开运算
4. 黑帽 = 闭运算 - 原图
5. 基本梯度 = 膨胀 - 腐蚀
6. 创建跟踪条并将其附加到指定的窗口：

createTrackbar(const String& trackbarname,const String &winname,int \*value,

int count，TrackbarCallback onChange = 0,void \* userdata = 0 )

创建拉伸条：createTrackbar("num:", "output", &element, max\_size, CallbackFunc);

\* 建模中的类图；

[拉普拉斯](https://so.csdn.net/so/search?q=%E6%8B%89%E6%99%AE%E6%8B%89%E6%96%AF&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_44753371/article/details/_blank)金字塔：图像的上采样图像的宽高会增加到原来的2倍：pyrUp(src, dst, Size(src.cols \* 2, src.rows \* 2));

高斯金字塔：图像的降采样(下采样) --图像的宽高减少到原图的1/2倍。

pyrDown(src, dst, Size(src.cols /2, src.rows /2));

高斯不同的意思就是对于同一张图像，采用不同的参数时，得到不同的灰度图进行相减，结果图像就是高斯不同。主要应用于灰度图的增强，以及角点检测。

步骤：

- 1、进行灰度转换

- 2、高斯模糊g1

- 3、高斯模糊g2

- 4 、相减：g1-g2

- 5、归一化：最大最小值归一化

//归一化 normalize(result, result, 255, 0, NORM\_MINMAX); 获取图片信息的轮廓；

6、阈值：double cv::threshold (InputArray src, OutputArray dst, double thresh,

double maxval, int type)

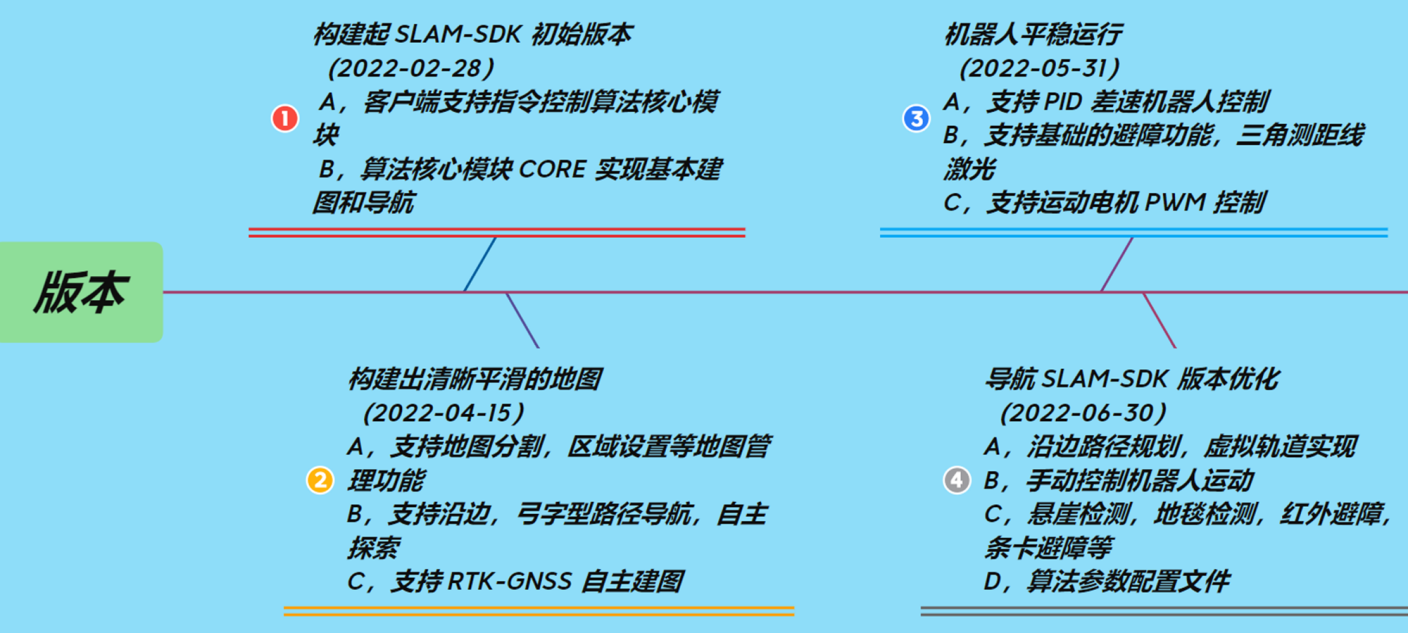
参数一：输入图像： 参数二：输出图像； 参数三：设置的阈值；参数四：阈值的最大值；

参数五：阈值类型；

高斯不同 + 归一化 + 阈值处理 可以得到图像内容的边缘信息；

图像直方图：是指对整个图像在灰度范围内的像素值（0 ~255）统计出现的频率的次数，成为图像直方图，直方图反映了图像灰度的分布情况，是图像的统计学特征；

2022/1/24：2021年优必选工作总结：

1. 搭建开发环境：
   1. 搭建 Linux 开发环境 "VMware Workstation 16 player" + "Ubuntu 18.04"；
   2. 安装ROS(melodic) 开发环境，包含 rviz，gazebo 等工具；
   3. 学习ROS的基础知识，通信机制、节点程序、launch文件等；
2. 熟悉 ROS 占据栅格地图的数据结构 OccupancyGrid格式：
   1. 熟练掌握导航栅格地图结构，以及对地图数据进行优化；
   2. 方法一：PGM图片采用二进制数据的方式保存，从数据的角度出发，难度较大，没找的合适的方法；
   3. 方法二：使用图像处理的方法对PGM图片进行优化，安装OPENCV-4，学习图像处理的基础知识，参考网址：https://blog.csdn.net/Current\_River/article/details/107338474
   4. 利用OPENCV的基础API接口函数，对PGM栅格地图进行优化（边缘不清晰栅格进行优化），并保存图片：
3. 首先读取图片，采用灰度格式读取；
4. 利用阈值将图片的灰度等级划分为三个等级：黑、白和灰；
5. 使用滤波API函数：将图片中的噪点滤掉；
6. 使用腐蚀API函数：将图片中的缝隙进行弥补，图片的信息轮廓变大；
7. 使用膨胀API函数：将图片中的信息轮廓变小，且缝隙连接上；
8. 最后将图片输出保存为PGM格式图片；
9. 梳理出OPENCV 依赖库，且编译OPENCV源码；
10. 对 Slam 建图有未知区域进行优化：
    1. 从激光雷达扫描的算法出发对图片进行优化；
    2. 优化扫描算法；
11. 掌握机器人地图管理知识，并实现地图（房间）切割，地图描述id，虚拟墙设计，面积计算（未知区域，障碍区域，空闲区域）计算：
12. 利用vornoni房间分割方法对栅格图片进行房间分割，
13. 根据分割后各房间的颜色不同，记录各个房间占据的像素，根据像素计算出房间的面积：像素个数 \* 分辨率 \* 分辨率 = 房间面积（）；
14. 对房间进行编号（设置id）变成单独一片区域；选择相邻房间合并（在APP端实现）
15. 框选要清扫的区域，框选禁止区域，画线选定禁止区域；（在APP端实现）
16. 了解机器人导航功能，对系统有初步的了解：  
    

2022/1/25：新员工自我评价：

入职的第一个月，由于工作内容和环境的变化，我花费了一些时间去重新适应新的集体，从公司的企业文化、发展理念、团队精神和工作内容等方面进行全面的了解，使我在今后的工作中更好的融入集体；

工作内容：开始学习机器人导航这个方向，具体任务是完成机器人导航栅格地图的优化，根据试用期工作和发展目标安排，第一个月完成的工作：

1、搭建开发环境：搭建 Linux 开发环境 "VMware Workstation 16 player" + "Ubuntu 18.04"；

安装ROS(melodic) 开发环境，包含 rviz，gazebo 等工具；

学习ROS的基础知识，通信机制、节点程序、launch文件等；

 2、熟悉 ROS 占据栅格地图的数据结构 OccupancyGrid格式；

3、对导航栅格地图数据优化提出解决方案：

方案一：PGM图片采用二进制数据的方式保存，从数据的角度出发，难度较大，没找的合适的方法；

方案二：使用图像处理的方法对PGM图片进行优化，安装OPENCV-4，学习图像处理的基础知识，利用OPENCV的基础API接口函数，对PGM栅格地图进行优化（边缘不清晰栅格进行优化），并保存图片；

总结：在今后的工作中更好的融入集体，加强自己的团队精神，在工作中严格要求自己，认真完成工作任务，提升自己的工作能力和综合素质。

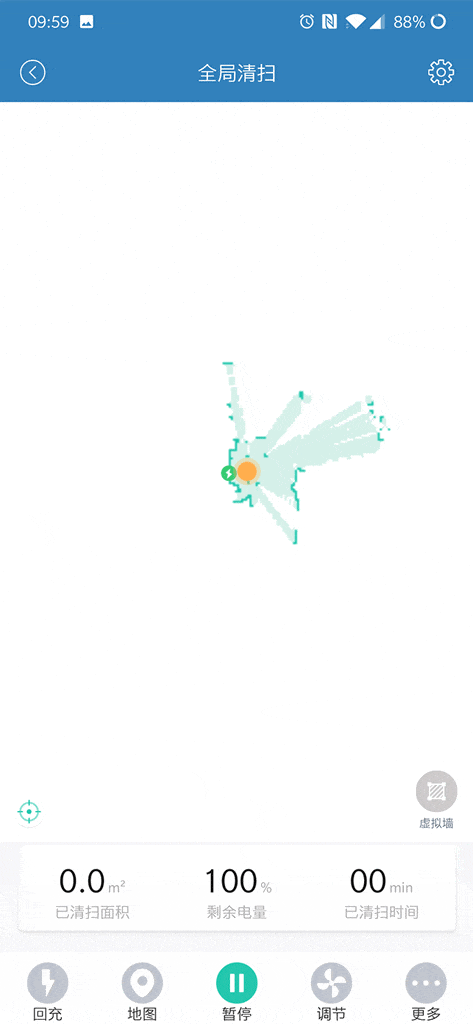
查找资料：

虚拟墙：网址：<https://baijiahao.baidu.com/s?id=1645539798768852205>

网址：<https://www.smzdm.com/zy/detail/av6plqm/>

网址：<https://www.bilibili.com/read/cv836334>

<http://www.cfan.com.cn/2019/0227/132006.shtml>



2022/1/26：每月谈话：

1. 工作内容是否明确？栅格地图优化，从图像处理的方法去做
2. 感觉团队怎样？实习期任务是否太重：
3. 要有明确的结果，要学会展示结果；
4. 连接远程服务器的方法：https://www.jianshu.com/p/14027e35b900
5. 利用放假的时间系统的学习一下图像处理的基础知识；