

# 具有无线充电的变电站巡检机器人应用软件

# 目录

|                       |    |
|-----------------------|----|
| 一、 Web 端机器人管理软件 ..... | 1  |
| 1.1 机器人管理.....        | 1  |
| 1.2 任务管理.....         | 2  |
| 1.2.1 全面巡检.....       | 2  |
| 1.2.2 专项巡检.....       | 2  |
| 1.2.3 例行巡检.....       | 3  |
| 1.2.4 任务展示.....       | 3  |
| 1.2.5 自定义任务.....      | 4  |
| 1.3 实时监控.....         | 5  |
| 1.4 巡检结果.....         | 5  |
| 1.5 巡检结果分析.....       | 6  |
| 1.6 机器人系统调试维护.....    | 6  |
| 二、 识别软件.....          | 6  |
| 2.1 开发工具.....         | 6  |
| 2.2 表计读数自动读取.....     | 7  |
| 2.2.1 主变油位识别需求分析..... | 7  |
| 2.2.2 避雷器监视器.....     | 14 |
| 2.2.3 断路器仪表.....      | 30 |
| 2.3 红外测温.....         | 33 |
| 2.3.1 绝缘子测温.....      | 33 |
| 2.3.2 测试结果.....       | 38 |
| 三、 导航系统软件.....        | 38 |
| 3.1 使用环境与配置.....      | 38 |
| 3.1.1 开机启动配置.....     | 38 |
| 3.1.2 配置上位机.....      | 39 |
| 3.2 在线建图步骤.....       | 40 |
| 3.3 地图标注步骤.....       | 42 |
| 3.3.1 基本显示功能.....     | 42 |
| 3.3.2 生成新的导航地图.....   | 43 |
| 3.3.3 标注导航地图.....     | 44 |
| 3.3.4 添加任务和导航点.....   | 45 |
| 3.4 执行任务操作步骤.....     | 46 |
| 3.4.1 基本显示功能.....     | 46 |
| 3.4.2 任务显示.....       | 46 |
| 3.4.3 发送任务点.....      | 46 |
| 3.4.4 硬件操作.....       | 47 |
| 3.4.5 任务信息显示.....     | 48 |
| 3.4.6 远程发送文件.....     | 49 |
| 3.4.7 远程获取文件.....     | 49 |
| 四、 结束语.....           | 50 |

## 一、Web 端机器人管理软件

Web 端机器人管理软件系统主要分为 6 个模块：机器人管理、任务管理、实时监控、巡检结果、巡检结果分析、机器人系统调试维护和用户设置。机器人管理模块包括实时信息，设备告警信息，系统告警信息，机器人基本信息查看等功能。任务管理模块包含全面巡检、例行巡检、专项巡检，特殊巡检、自定义任务、地图选点、任务展示，并且任务具有查询，保存重置，导入，删除等功能，并能够对各种巡检任务进行的精确化管理。实时监控模块包含巡检视频、红外监控，可以方便地进行监控管理。巡检结果模块具有设备告警信息、巡检结果浏览、巡检报告生成等功能，可以方便对巡检结果进行操作管理。巡检结果分析模块提供了对比分析、生成报表等功能。用户设置模块用于对告警设置、权限设置、点位设置、检修区域设置等系统级操作。机器人系统调试维护模块包含了巡检地图设置、软件设置、机器人设置、机器人信息查询等功能。

### 1.1 机器人管理

机器人管理模块主要为了实现一站多台机器人之间的选择切换，子功能模块包括站内所有机器人。选择其中一个机器人后，其中的任务管理、实时监控（除巡检报文）二大模块均切换至对应的机器人工作界面。

功能：机器人查询，实时信息，设备告警信息，系统告警信息等  
界面如下：



图 1.1 机器人管理界面

## 1.2 任务管理

任务管理模块主要包括全面巡检、例行巡检、专项巡检、自定义任务及任务展示等子功能模块。专项巡检包括红外测温、油位/油温表抄录、避雷器表计抄录、SF6 压力表抄录、液压表抄录。每个子功能模块集合了任务编制、任务下发等功能。根据不同巡检任务类型自动预设相关巡检点位，并自动生成任务名称。

任务下发主要提供三种方式供用户选择：

- 1) 立即执行：当前时间立即开始本次巡检任务，并暂停正在执行的任务。
- 2) 定期执行：从日历选择巡检日期，确定时间，按要求执行巡检任务。
- 3) 周期执行：可选择每日、每周、每月、固定间隔天数、从日历日期选择（多选）方式，确定时间，按要求周期性执行巡检任务。

### 1.2.1 全面巡检

全面巡检是指通过预先设定站内设备的表计、状态指示、接头温度、外观及辅助设施外观、变电站运行环境等巡检点，快速生成抄录任务。

功能：从设备区域，设备类型，识别类型，表计类型，设备外观检查方面对变电站下设备在全面巡视下的查询，保存，重置功能。

### 1.2.2 专项巡检

专项巡检是指对巡检站内特定设备的表计、状态指示、接头温度、外观及辅助设施外观、变电站运行环境等巡检点，快速生成抄录任务。

功能：从设备区域，设备类型，识别类型，表记类型，设备外观检查方面对变电站下设备在全面巡视下的查询，保存，重置功能。

### 1.2.3 例行巡检

例行巡检是指预先设定站内设备的表计、状态指示、外观及辅助设施外观、变电站运行环境等巡检点，快速生成抄录任务。

功能：从设备区域，设备类型，识别类型，表记类型，设备外观检查方面对变电站下设备在例行巡视下的查询，保存，重置功能。

巡检界面如下图

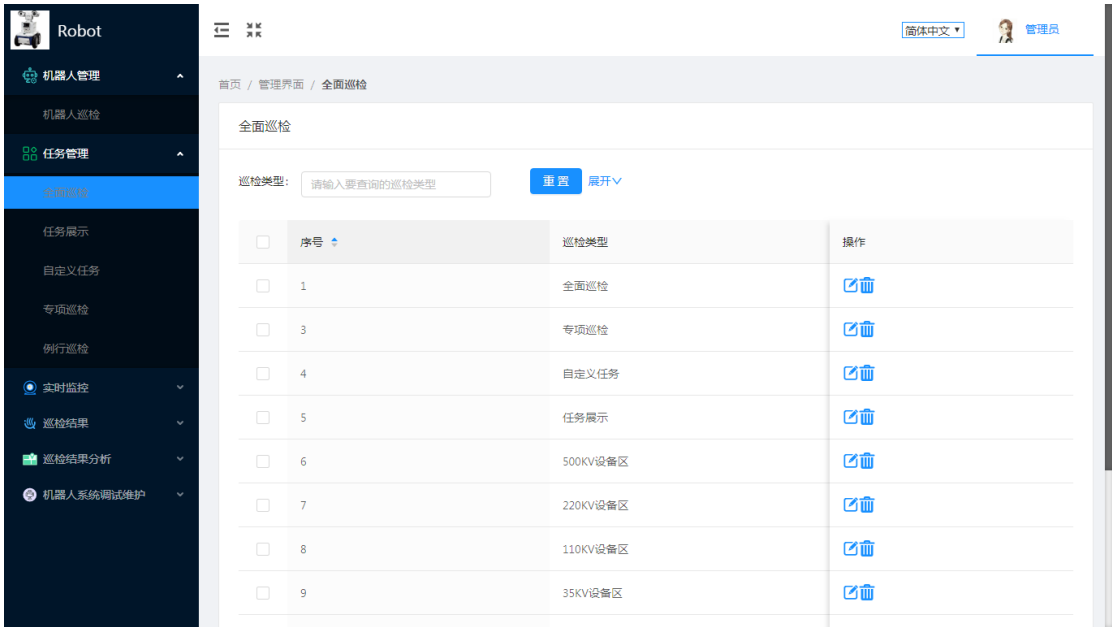


图 1.2 巡检界面

### 1.2.4 任务展示

任务展示显示所有下发的巡检任务。包括所有已下发任务的执行时间、任务名称。

窗口上方提供按序号、任务名称关键字查询功能。

- 1) 添加：用户点击后链接至任务编制窗口
- 2) 修改：用户点击当前任务后链接至任务编制窗口进行修改，同时提供返回按钮。

3) 删除：删除用户选中任务。

任务展示界面：



图 1.3 任务展示界面

### 1.2.5 自定义任务

自定义任务主要实现新任务编制功能。任务编制可以通过两种方式：历史任务导入、自定义编制。用户点击自定义任务功能菜单后，首先显示自定义任务编制窗口，其上方提供历史任务导入按钮。用户选择任一历史任务导入后，均可以通过自定义编制进行修改。两种方式的详细功能分别描述如下：

#### (1) 历史任务导入

用户单击历史任务导入后，窗口右侧弹出已完成的历史任务窗口，窗口上方提供按任务完成时间、任务名称关键词查询功能。窗口中部以列表形式显示相应查询条件下的历史任务。

#### (2) 自定义编制

该窗口显示内容分上中下三块区域，上方为任务必填字段输入区域。分别为：任务名称：确定方式为变电站+自定义任务描述+系统时间（年月日）；自定义任务描述：由系统根据用户选择的筛选条件、巡检点信息自动生成，允许用户进行修改编辑。中间区域提供各类筛选条件供用户选择，分别为：巡检类型、设备区域、设备类型、识别类型、表计类型、设备外观类型。

自定义任务编制功能按钮如下：

1) 保存：保存当前任务。

2) 重置：将所有选择巡检点位清空，待用户重新设置。

功能：从设备区域，设备类型，识别类型，表计类型，设备外观检查方面对变电站下设备在例行巡视（自定义任务）下的查询，保存，重置功能。

自定义任务界面如下：

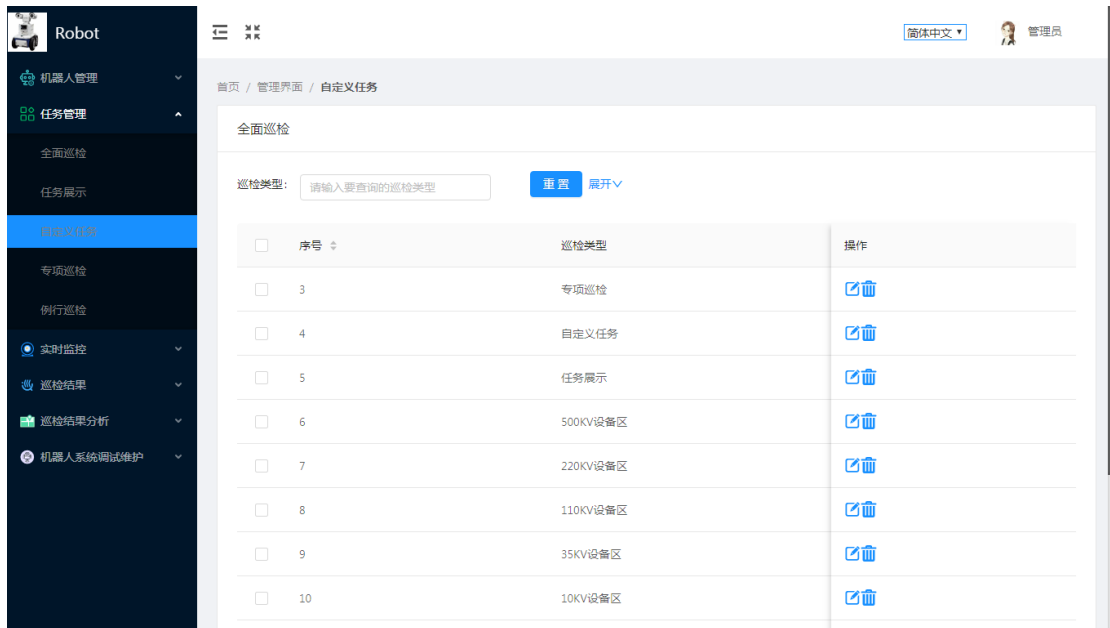


图 1.4 自定义任务界面

### 1.3 实时监控

实时监控是包括可见光视频、红外视频。

#### 1) 可见光视频

可见光视频直观展示机器人当前巡检设备的实时图像，显示当前云台水平位置、垂直位置，可单图全屏放大。主要包括表计、位置状态、设备外观等。

#### 2) 红外视频

红外视频直观展示巡检机器人当前巡检设备的红外图像，可单图全屏放大。主要包括设备温度、环境温度等。

### 1.4 巡检结果

巡检结果确认模块主要包括设备告警确认、巡检结果浏览、巡检报告生成三个模块。用于对机器人的巡检结果，包括各类设备告警信息数据，以及各巡检点位采集的图像、音视频信息等进行核查确认，并生成巡检报告。

巡检结果界面如下：



图 1.5 巡检结果界面

## 1.5 巡检结果分析

巡检结果分析实现各巡检点位任务的查询、浏览、输出功能，分析各设备的巡检覆盖情况，并可实现对该巡检任务信息查询、浏览及审核确认，具备导出相应报告。

## 1.6 机器人系统调试维护

机器人系统调试维护模块主要实现巡检地图维护、软件设置及机器人设置等功能。机器人系统调试维护界面如下：



图 1.6 巡检结果界面

## 二、识别软件

### 2.1 开发工具

- 1) openCV 3.1.0
- 2) Visual studio 2017



## 2.2 表计读数自动读取

变电运行是供电企业的一个重要生产部门,变电站变电运行管理水平高低直接影响着电网运行的安全和稳定。为了确保电气设备的安全稳定运行,及时发现设备的缺陷或隐患,往往要求现场运维人员定期或不定期对现场设备进行巡视检查。这种人工模式,工作量大,且受环境因素、人员素质等各方面的影响,容易造成巡检不到位的情况发生,巡检效率和质量达不到预期的效果。

随着巡检机器人的逐步推广,变电所无人值班模式越来越收到电力公司的青睐。由于传统运行巡检获取的信息 95%以上来源于目视视觉,因此,在巡检无人机上配备红外成像和可见光相机,通过软件算法的开发,充当巡检机器的视觉。巡检机器人可以高频次、大范围、无遗漏的执行巡检任务,及时发现热故障、油位检测以及各种仪表数据的抄读。

在本系统开发中,所需要关注的是主变油枕油位、监视器数据抄读、断路器仪表抄读、隔离开关刀闸判别以及绝缘子红外测温。这些任务都交由红外图像处理系统和可见光图像处理系统,配合巡检机器人完成巡检效率。

在开发配套的图像处理系统中,首先需要对所要处理的场景进行建模,熟悉场景内所要监测的设备或仪表;以此开发所需要的功能。

### 2.2.1 主变油位识别需求分析

图 2.1 为主变油位的现场图。目标是对当前油位进行判别,当超过标识的上限或低于下限,则需要反馈给后台进行报警;当油位处于正常范围时,需要对油位进行判读。



图 2.1 主变油枕油位识别

#### 1) 油位特征描述

由图 1 可知，油枕油位由红色条纹进行标记，且油位的上限和下限以及正常油位的百分比都是由红色区域标定。

#### 2) 油位特征提取

##### a) 颜色建模

在可见光图像中，通常由红(R)、绿(G)、蓝(B)三色构成。这三种颜色的不同组合可以形成几乎所有的其他颜色。而在自然环境下获取的图像容易受自然光照、遮挡和阴影等情况的影响，即对亮度比较敏感。而 RGB 颜色空间的三个分量都与亮度密切相关，即只要亮度改变，三个分量都会随之相应地改变，而没有一种更直观的方式来表达。

为此，在本项目开发中，关于可见光图像处理部分，采用 HSV 颜色空间。HSV 表达彩色图像的方式由三个部分组成：Hue（色调、色相）、Saturation（饱和度、色彩纯净度）和 Value（明度）。

图 2.2 给出了 HSV 颜色空间的描述，横截面可以看做是一个极坐标系，H 用极坐标的极角表示，S 用极坐标的极轴长度表示，V 用圆柱中轴的高度表示。

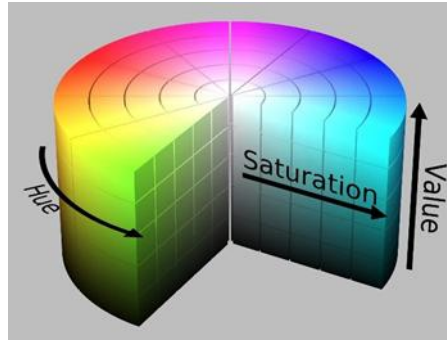


图 2.2 HSV 颜色空间描述

Hue 用角度度量，取值范围为  $0 \sim 360^\circ$ ，表示色彩信息，即所处的光谱颜色的位置。颜色圆环上所有的颜色都是光谱上的颜色，从红色开始按逆时针方向旋转，Hue=0 表示红色，Hue=120 表示绿色，Hue=240 表示蓝色等。

因此，在 HSV 颜色空间下，能够比 RGB 更容易跟踪某种颜色的物体，因此对于特定颜色的区域具有较好的效果。

在图像处理中，关于 RGB 图到 HSV 对应的映射关系为：

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases}$$

If  $H < 0$  then  $H \leftarrow H + 360$ . On output  $0 \leq V \leq 1$ ,  $0 \leq S \leq 1$ ,  $0 \leq H \leq 360$ .

具体关于颜色的选择可参考表 2.1。

|      | 黑   | 灰   | 白   | 红   |     | 橙   | 黄   | 绿   | 青   | 蓝   | 紫   |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| hmin | 0   | 0   | 0   | 0   | 156 | 11  | 26  | 35  | 78  | 100 | 125 |
| hmax | 180 | 180 | 180 | 10  | 180 | 25  | 34  | 77  | 99  | 124 | 155 |
| smin | 0   | 0   | 0   | 43  |     | 43  | 43  | 43  | 43  | 43  | 43  |
| smax | 255 | 43  | 30  | 255 |     | 255 | 255 | 255 | 255 | 255 | 255 |
| vmin | 0   | 46  | 221 | 46  |     | 46  | 46  | 46  | 46  | 46  | 46  |
| vmax | 46  | 220 | 255 | 255 |     | 255 | 255 | 255 | 255 | 255 | 255 |

表 2.1 HSV 空间中颜色对应表

#### b) 读数建模

从主变油枕油位可以看出，上限、下限以及各个标记均与显示窗具有一定的关系。为了避免可见光摄像头的焦距的变化，因此，需要建立各个标记油位  $h_i$  占总显示窗高度  $h$  的比例关系， $i=0, 1, 2, 3, 4, 5$  分别代表下 0%，20%，40%，60%，80%，上限。因此，当检测到在图像中高度  $h'$  以及各个标记为的高度  $h_i'$ 。其中

测量值  $h_i'$  目的是为了更高精度测量。在要求不高的情况下，油位  $h_r$  的百分比可按式进行计算：

$$\frac{\left(h_r' - \frac{h_0}{h} h'\right)}{\left(h' - \frac{h_0}{h} h'\right)} \times 100\%$$

### 3) 目标匹配

在图像中，由于外界的干扰，仅采用颜色检测标记，容易引发错误。为了能够准确的对准目标，本处理采用 Harris 角点方法。

Harris 是 Harris 和 Stephens 在 1988 年提出，是定义在各个方向上灰度值变化的点。与 Moravec 算子的不同之处在于局部自相关测量结果的估计方式，Moravec 算子只对八方向离散的移动方向计算灰度变化，而 Harris 算子允许获得所有方向上的自相关变化，也就是灰度，它用一个自相关函数来计算灰度信号在二维方向上具有明显变化的像素点位置，构造一个与相关函数相关的矩阵  $M$ ，而通过比较矩阵  $M$  的特征值的大小可以轻松地提取到相应的角点。

设图像窗口平移  $[u, v]$ ，产生的灰度变化为  $E[u, v]$ ，则

$$E[u, v] = \sum_{x, y} \omega(x, y) [I(x+u, y+v) - I(x, y)]^2$$

由  $I(x+u, y+v) = I(x, y) + I_x u + I_y v + o(u^2, v^2)$  得到

$$E[u, v] = \sum_{x, y} \omega(x, y) [I_x u + I_y v + o(u^2, v^2)]^2$$

$$\text{考虑到 } [I_x u + I_y v]^2 = [u, v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

于是，对于局部微小的移动量  $[u, v]$ ，可以近似得到下面的表达

$$E[u, v] \approx [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M \text{ 为二维矩阵，且 } M = \sum_{x, y} \omega(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$\text{其中 } \omega(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}$$

其中  $I_x$  标识  $x$  方向的梯度,  $I_y$  表示  $y$  方向的梯度, 为高斯函数。矩阵  $M$  的特征值是自相关函数的一阶曲率。 $M$  特征值的大小与特征点的性质息息相关。即当两个特征值都比较小时, 则此点可能位于平坦区, 不为角点或边界点。当两个特征值均相对较大时, 则此点沿任意方向的曲率都较大, 为需要提取的角点。

一般地, 为了避免计算矩阵  $M$  的特征值, 常通过下式来计算特征点的响应函数, 亦称兴趣值

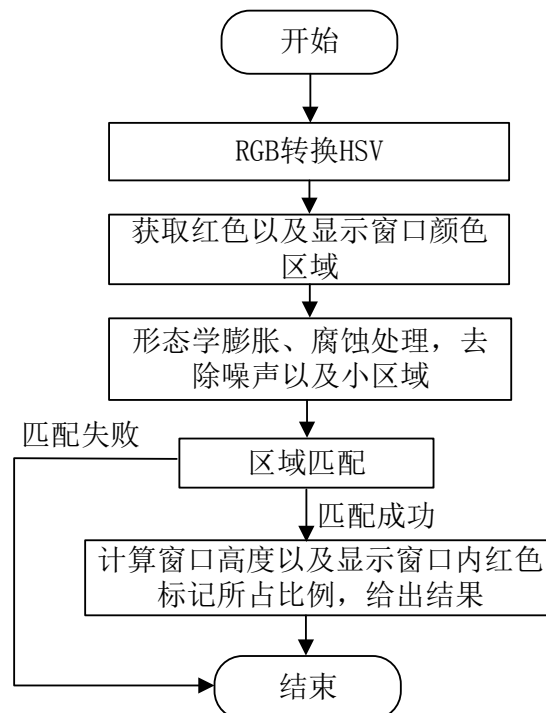
$$R = \text{Det}(M) - K * \text{Trace}^2(M)$$

其中  $\text{Det}(M)$  为矩阵  $M$  的行列式,  $\text{Trace}(M)$  为矩阵  $M$  的迹, 即  $M$  特征值之和;  $K$  为经验值常数, 一般在  $0.04 \sim 0.06$  范围内取值。为了计算方便, 采用如下角点响应函数计算兴趣值从而判定出角点, 即

$$R = \left[ I_x^2 * I_y^2 - (I_x I_y)^2 \right] / (I_x^2 + I_y^2 + \varepsilon)$$

其中,  $\varepsilon$  表示任意小的正数。显然采用这中方式计算可以避免  $K$  选择的随机性。

#### 4) 油位读数算法流程

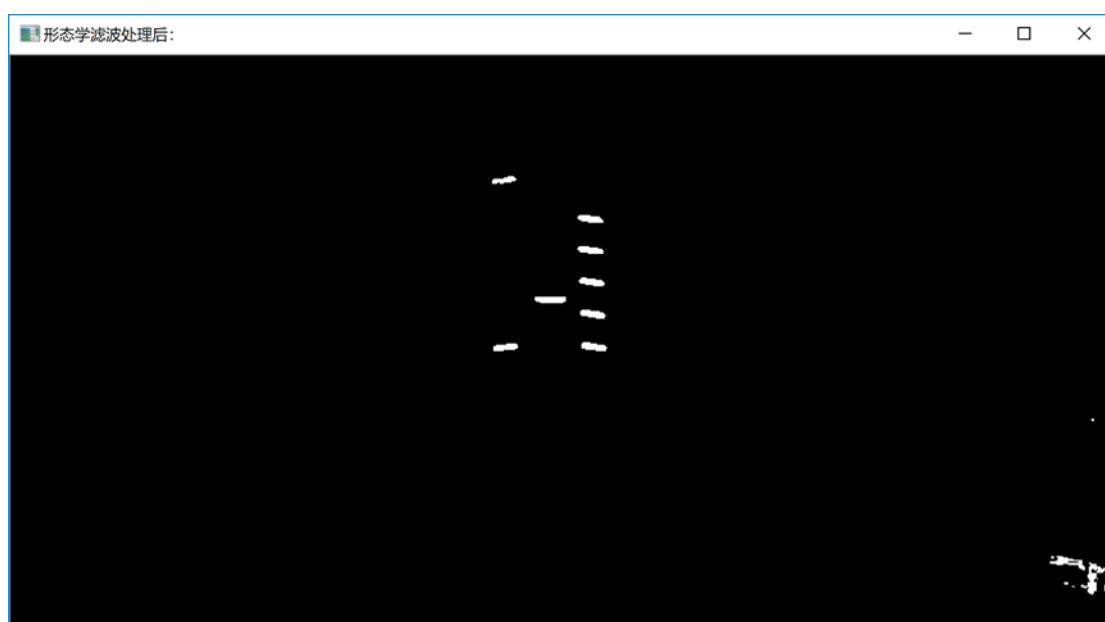


#### 5) 读数结果测试

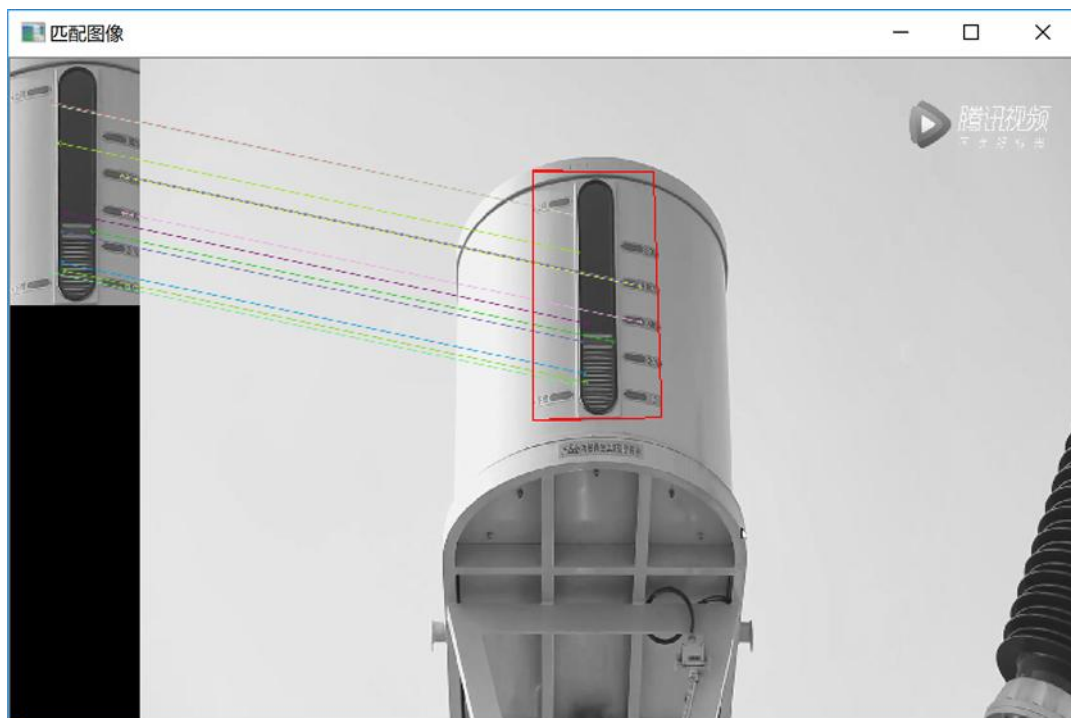
第一步: 区域提取



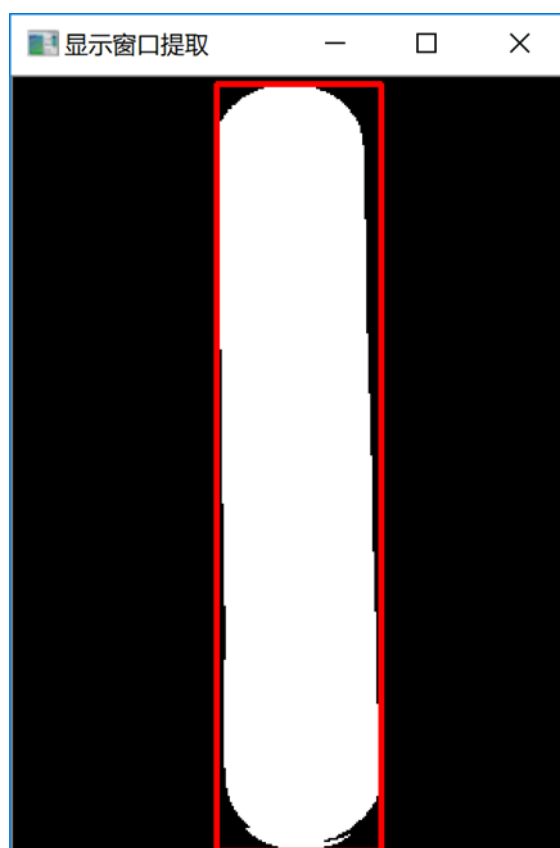
第二步，形态学滤波，去除小区域



第三步，区域匹配



第四步，提取显示窗口



计算红色油位标记区域的中线与整个显示窗的比例计算，得到的结果值  
0.2889, 即 28.89%

### 2.2.2 避雷器监视器

下图为一种半弧形指针式仪表，通常相对较小，而且其读数的复杂程度要高于油位这种方式的检测。同样地，对于指针的区域由绿、黄、红三色指定仪表所在的区域。为了能够准确的获取读数，则必须对指针进行检测，通过他在各个区域内的比例，确定最终的度数。



#### 1) 表盘匹配

在巡检机器人巡检过程中，由于仪表通常较小，需要对可见光摄像头的焦距进行调整。因此，需要首先对这个目标进行定位或者匹配。

目前，图像匹配技术主要是基于特征点的两种匹配方式。Harris 算法是一种改进的 Moravec 算法，该算法利用高斯核函数来代替 Moravec 算法中的矩形窗口，从而使得特征点对于图像各种变换具有更强的鲁棒性。同时，Harris 定义了一种角点响应函数来改善 Moravec 算法的角点边缘效应。然而，当对角点的兴趣值进行非极大值抑制来确定局部极大值的时候，角点的提取效果几乎完全由设定的阈值大小决定。而阈值的大小也与所提取的角点数量息息相关，一般情况下，阈值越大提取的角点越少，极易造成正确角点的丢失；阈值越小提取的角点数越多，也会带来很多伪角点。因此，在运用 Harris 算法进行角点检测时，阈值这个经验值的选取和设定对角点提取具有很大的影响。

SIFT (Scale Invariant Feature Transform) 算法的出现解决了这个问题。SIFT 算法在 1999 年被 Lower 等人首次提出。该算法利用高斯核函数构建图像的多尺度空间，有效地解决了特征点对图像尺度不变性的问题，成为了目前最为有



效、应用也最为广泛的图像特征点检测算法。进一步地，Bay 等人在 2006 年提出的 SURF (Speed Up Robust Features) 算法，是对 SIFT 算法的一种改进算法，该算法利用盒状滤波器对 SIFT 算法中的高斯核函数进行近似，并使用了积分图像与快速 Hessian 矩阵简化卷积运算的过程。因此，SURF 算法的运算速度比 SIFT 算法提高了很多。

#### a) SURF 特征点提取

SURF 算法运用了积分图像的概念，加快了图像特征点的提取时间。SURF 算法在特征点提取部分主要分为三个步骤：建立高斯尺度空间、基于 Hessian 矩阵的特征点检测、特征点的精确定位，其中最后一步的原理与 SIFT 算法相同，即非极大值抑制和插值原理。

#### ● 积分图像的概念与算法

积分图像指在原始图像中，原点与点  $(x, y)$  所构成矩形区域内，所有像素点的灰度值之和就是该点的积分图像值。定义如下：

$$I_{\Sigma}(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)$$

如图 2.3 所示，某图像中有四个点，坐标分别为  $A(x_0, y_0)$ 、 $B(x_1, y_0)$ 、 $C(x_0, y_1)$ 、 $D(x_1, y_1)$ 。由这 4 个点组成了矩阵，该矩阵内的像素灰度之和为：

$$\sum_{\substack{x_0 < x \leq x_1 \\ y_0 < y \leq y_1}} I(x, y) = I_{\Sigma}(A) + I_{\Sigma}(D) - I_{\Sigma}(B) - I_{\Sigma}(C)$$

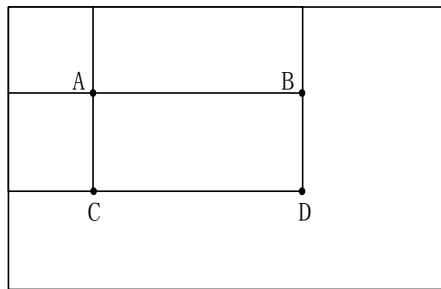


图 2.3 积分图像求矩阵内灰度值

SURF 算法采用积分图像，能够快速地完成不同尺寸大小盒状滤波器与原始图像的卷积过程，使得图像特征点的提取效率显著提高。

#### ● 尺度空间的建立

如果想要在缩放关系的图像中寻找最佳的匹配特征点，检测算法就必须具

备在不同尺度空间下找到同一特征点的能力。因此如何建立连续多幅的尺度图像就成为特征点检测的关键，在现有的特征点检测算法中是需要建立图像尺度金字塔，如 SIFT 算法，在上一小节已经阐述，这里将不再叙述。

在 SURF 算法中，利用盒子滤波器来建立图像的多尺度空间，通过改变盒子滤波器的尺寸大小，从而得到不同尺度下的图像。在 SURF 算法建立多尺度的方法，可以具体分为以下 3 个步骤：

#### ◆ 构建基础尺寸盒状滤波器

如图 2.4 所示，第一行图像就是经过离散化处理的高斯核函数，其尺度  $\sigma=1.2$ ，并被裁剪为  $9\times 9$  的方格。在 SURF 算法中，最小的空间尺度为 1.2，同时也是空间的最高分辨率。

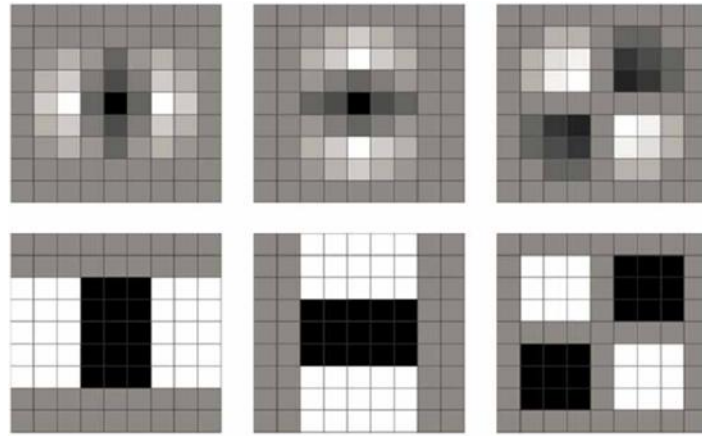


图 2.4 高斯二阶微分算子和盒子滤波器

第一行图像从左至右，分别代表沿  $x$  方向、 $y$  方向和  $xy$  方向的高斯二阶微分算子。而如图 2.4 第二行图像所示的  $9\times 9$  盒子滤波器 (boxfilters)，从左至右依次为  $D_{xx}$  模板、 $D_{yy}$  模板和  $D_{xy}$  模板，则是对上述三种微分算子的近似。

在盒子滤波器中，白色和黑色部分统称为突起部分，在  $D_{xx}$  模板和  $D_{yy}$  模板中，他们所对应的权值分别为 1 与 -2，而在  $D_{xy}$  模板中，黑色突起部分的权值为 -1。而灰色部分称为平坦部分，其所对应的权值为 0。

在 SURF 算法的图像多尺度空间中，各组各层图像均由不同尺寸大小的盒子滤波器与原始图像做卷积而得到的，而不同尺寸大小的盒子滤波器则是通过基础尺寸的盒子滤波器扩展得到的。

#### ◆ 对基础尺寸的盒子滤波器的扩展

对基础尺寸的盒子滤波器进行扩展,可以得到图像中不同尺寸的盒子滤波器,这样就可以用不同尺寸的盒子滤波器替代不同尺度因子的高斯核,从而建立多尺度空间下连续的尺度函数。

如图 2.5 所示, SURF 算法第一组各层图像的盒子滤波器尺寸分别为  $9 \times 9$ 、 $15 \times 15$ 、 $21 \times 21$ 、 $27 \times 27$ , 按照最小步长 6 的方式扩展; 图像堆的第二组中模板扩展步长变为 12, 模板大小分别为  $15 \times 15$ 、 $27 \times 27$ 、 $39 \times 39$ 、 $51 \times 51$ ; 第三组第四组的步长分别翻倍至 24、48 模板尺寸以此类推。其中, 各组各层图像所对应的盒子滤波器尺寸计算方法为

$$s_{approx} = L \times \frac{s_0}{L_0} = L \times \frac{1.2}{9}$$

式中  $L_0$  表示基准盒子滤波器模板的尺寸,  $s_0$  表示盒子滤波器模板所对应的空间尺度, 而  $L$  则是当前盒子滤波器模板的尺寸。

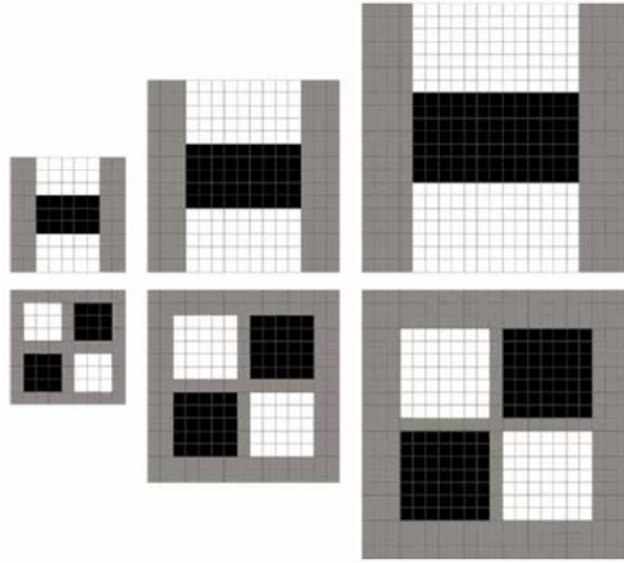


图 2.5 盒子滤波器模板的扩展

SURF 算法的图像多尺度空间中, 各组图像所对应的盒子滤波器尺寸之间是有重叠的, 这样做目的是尽可能覆盖所有的尺度。随着图像尺度的增加, 所检测出的特征点数目也会随之减少。所以, 一般采用隔点降采样的方式来提高图像特征点的检测速度。

#### ◆ 使用不同尺度的盒子滤波器构建多尺度空间

分别用上述十六个不同尺度的盒子滤波器与原始图像进行卷积运算, 得到了十六个不同尺度下的灰度图像, 把这十六个灰度图像组合就成为了一个连续的多

尺度空间。

- 基于 Hessian 矩阵的特征点检测

通过计算图像上所有像素点 Hessian 矩阵的行列式值，就可以判断哪些像素点为图像的特征点。而 SURF 算法利用快速的 Hessian 矩阵行列式值的计算方法，也称 Fast-Hessian 算法。该算法的具体原理如下：

对于图像  $I(x, y)$  中的任意一个像素点  $(x, y)$ ，Hessian 矩阵如下：

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}$$

式中， $L_{xx}(x, y, \sigma)$  为高斯核函数在点  $(x, y)$  处对  $x$  方向的二阶偏导  $\partial^2 G / \partial x^2$  与图像  $I(x, y)$  的卷积。而  $L_{xy}(x, y, \sigma)$ 、 $L_{yy}(x, y, \sigma)$  含义相同。 $\sigma$  为该像素点所对应的空间尺度。而 SURF 算法正是通过盒子滤波器模板与积分图像来计算图像中任意像素点的 Hessian 矩阵。

利用公式将图像转换为积分图像，然后用盒子滤波器模板来计算图像中任意像素点 Hessian 矩阵的行列式值。从图 2.4 中看出，盒子滤波器中的灰色部分仅仅充当模板填充作用，权值为 0。 $D_{xx}$  和  $D_{yy}$  模板中有三个突起部分，均是两个白色部分和一个黑色部分，而  $D_{xy}$  模板有四个突起，分别是两个白色部分和两个黑色部分。

因此，利用盒子滤波器对图像进行滤波处理所得到的响应值的公式为：

$$D = \sum_{n=1}^N \frac{W_n}{S_n} (I_{\Sigma}(A_n) + I_{\Sigma}(D_n) - I_{\Sigma}(B_n) - I_{\Sigma}(C_n))$$

公式中  $N$  表示突起部分的总和； $D_{xx}$  和  $D_{yy}$  模板的  $N$  为 3 而  $D_{xy}$  模板的  $N$  为 4； $S_n$  表示突起部分的面积，之所以除以  $S_n$  的意义在于将其归一化处理； $W_n$  代表着第  $n$  个突起部分的权值，求模板的每个突起部分对应于图像中 A、B、C、D 所组成的矩阵区域的灰度值之和也就利用上了前面介绍的积分图像。

然后，利用盒子滤波器与原始图像的卷积结果来近似代替高斯二阶微分算子，从而得到 Hessian 矩阵的行列式值。具体计算方法为

$$\det(H_{approx}) = D_{xx} D_{yy} - (D_{xy})^2$$

$w$  是权重系数，是用来平衡因为近似替代高斯二阶微分算子所带来的偏差，一般在实验中取 0.9。

SURF 算法在特征点定位阶段与 SIFT 算法一致，在这里将不再重述。

#### b) 特征描述子

图像特征点的描述是计算机视觉领域中图像特征点匹配的一个基本组成环节，其核心思想是要获得一个特征描述向量来代表图像特征点及其周围领域的信息。这个特征描述向量也就是图像特征点的描述子，之后的图像特征点匹配工作就是依据这些局部特征描述子，使用某种相似性度量方法，从而找到特征点之间的匹配关系，完成图像的匹配。

传统的 SIFT 算法在得到图像特征点的位置以及尺度参数后，利用图像特征点在其周围领域内的梯度方向图对特征点进行描述，可得到一种基于梯度的特征点描述子。

目前，基于梯度直方图的描述子是应用最广泛的图像特征点描述方法之一，该方法通过计算特征点局部方位内的梯度信息，并且对该范围内的梯度方向以及梯度的大小进行统计，最后将这些信息组合起来就可以得到对应的特征点的特征描述子，该方法主要包括特征点主方向的确定和特征点描述子的生成两个步骤。

在本节中，将对基于梯度直方图的描述子中的 SURF 特征点描述子进行介绍。

#### c) SURF 特征点描述子

与 SIFT 算法不同的是 SURF 算法是通过引入 Haar 小波变换求特征点周围信息。与 SIFT 生成描述子过程相似，在生成 SURF 描述子的过程中也包含着两个步骤：特征点主方向的确定和根据特征点领域内的 Haar 小波响应信息来构建特征描述向量。下面将分别介绍这两个步骤。

##### (1) 特征点主方向的确定

在对特征点进行描述时，为了实现图像的旋转不变性，在提取图像特征点之后，就必须为每个特征点分配一个方向角度。如图 2.6 所示，首先，以特征点为圆心，半径为  $6s$  的圆领域进行采样，采样间隔为  $s$ ， $s$  是特征点所在尺度图像的尺度。

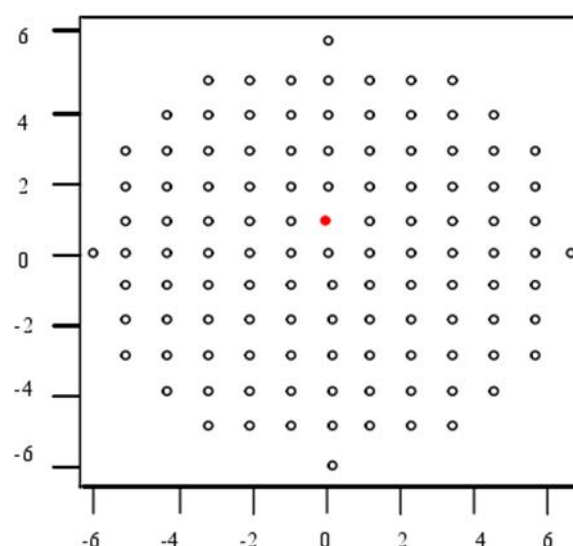


图 2.6 6s 圆领域

对于特征点周围 6s 圆领域内所有的像素点，利用 Haar 小波模板进行滤波。Haar 小波模板的尺寸是 4s。分别利用 x 方向与 y 方向的 Haar 小波模板求出水平和垂直方向的 Haar 小波响应值，并对这些像素点的响应值进行高斯加权。

Haar 小波模板如图 2.7 所示。其中左侧是 x 方向的 Haar 小波模板，右侧是 y 方向的 Haar 小波模板，白色部分的权值为-1，黑色部分为 1。之所以选择 Haar 小波的另外一个原因是可以运用积分图像，从而加快了运算速度。

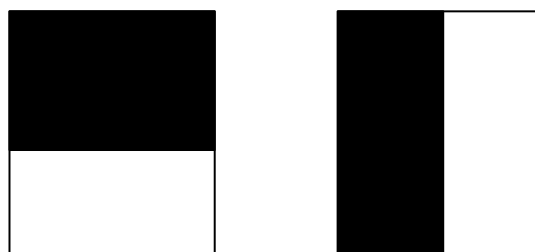


图 2.7 Haar 小波模板

之后，用加权后的 x 方向和 y 方向的 Haar 小波响应值作为 x 轴和 y 轴，建立一个坐标系，所有的 6s 圆领域内的采样点都分布在该坐标系上。求得特征点主方向的方法是：以坐标原点为中心，构造一个扇形，扇形的张角是  $60^\circ$ ，用此扇形，按顺时针的方向滑动 6s 圆领域，如图 2.8 所示，其中滑动步长一致，则该窗口内所有点的梯度和幅值为

$$m_w = \sum_k x + \sum_k y$$

$$\theta_w = \arctan\left(\frac{\sum_k x}{\sum_k y}\right)$$

式中， $m_w$  和  $\theta_w$  分别是扇形窗口内采样点的幅值和梯度， $k$  是窗口内所有像素点。

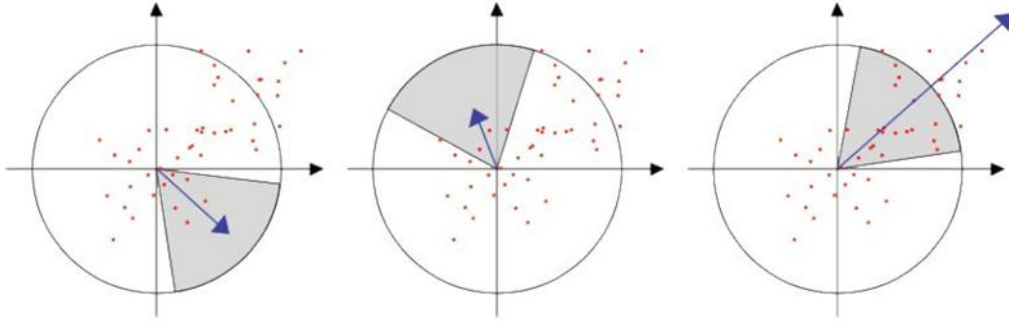


图 2.8 确定特征点主方向示意图

选择扇形滑动窗口，直至旋转一周为止，接着比较所有窗口的模值，其中模值最大的那个窗口对应的梯度就是该特征点的主方向，即

$$\theta = \theta_{w|\max(m_w)}$$

## (2) 特征点描述子的生成

首先，以特征点为中心构建  $20s$  的正方形领域，这里  $s$  仍然代表图像特征点所在图像多尺度空间中的图像尺度。在  $20s$  的正方形领域进行采样，并且把这个邻域划分为  $4 \times 4 = 16$  的子区域，这样在  $20s$  的正方形领域内共有 400 个采样像素点。这样每个小的区域有 25 个采样点，此处依旧采用 2sHaar 小波，分别计算  $x$ 、 $y$  方向的 Haar 小波响应值  $dx$ 、 $dy$ ，以特征点为中心，赋予这些像素点高斯权重系数。

然后，为了保持旋转不变性，正方形领域需要旋转到和特征点主方向一致的方向，如图 2.9 所示。

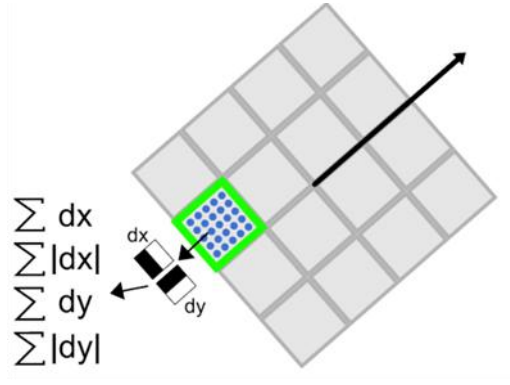


图 2.9 特征描述子表示

这样，在每个子区域中，累加 25 个采样点的  $dx$  和  $dy$ ，同时为了去除特征点的极性信息，也需要累加  $dx$ 、 $dy$  的绝对值，在每个子区域中也就形成了一个 4 维的特征描述向量即  $V$ ：

$$V = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$$

为了去除光照的影响，也需要对 SURF 描述子进行归一化处理，而归一化处理的具体方式与 SIFT 算法一致。

#### d) 特征匹配

相似性度量是一种计算两幅特征点之间相似度的方法，一般选取特征点描述子向量之间的欧式距离作为评价标准，如果两个特征点描述子向量之间的距离越小，那么这两个特征点就越匹配。常见的特征空间搜索策略主要有穷举以及 KD 树等搜索办法。其过程都是在两幅图像的特征空间中，以相同的相似性度量准则为依据，找寻最为相关的图像特征点对。目前，优化算法的飞速发展，为简化特征空间的搜索过程提供了帮助。

##### (1) 相似度量性准则

相似性度量准则是评价两幅图像的特征空间中两个描述向量相似度的标准，也是判断两个特征点是否匹配的唯一准则，在图像特征点的匹配过程中起到了至关重要的作用，为后续的匹配搜索过程提供依据。

因此，相似性度量准则是图像匹配的核心问题。在图像特征点的匹配算法中，一般采用特征点描述子向量的距离作为特征点相似性度量的准则，如欧氏距离等，是十分常见的相似性度量准则。

选取特征点描述子向量的欧氏距离作为特征点匹配过程中的相似性度量准



则，而欧氏距离的具体定义为

$$d_{ij} = \|a_i - b_j\| = \sqrt{\sum_{n=1}^L \sum_{m=1}^L (a_{in} - b_{jm})^2}$$

其中，向量  $a_i$  表示参考图像中第  $i$  个特征点的描述子向量，向量  $b_j$  表示待匹配图像中第  $j$  个特征点的描述子向量， $\|\cdot\|$  为向量的 2-范数，而  $d_{ij}$  则表示参考图像中第  $i$  个特征点与待匹配图像中第  $j$  个特征点之间的欧氏距离。而  $L$  为描述子向量的维度， $a_{in}$  则表示描述子向量  $a_i$  中的第  $n$  个元素。同样， $b_{jm}$  则表示描述子向量  $b_j$  中的第  $m$  个元素。

通过图像特征点描述子之间欧氏距离的计算，就可以得到两个特征点之间的相似度。特征点描述子之间的欧氏距离越大，代表这两个特征点的相似度越低；相反，特征点描述子之间的欧氏距离越小，则代表这两个特征点的相似度越高，也就是说这两个特征点越匹配。在图像特征点的匹配过程中，认为欧氏距离最小的一对图像特征点即为匹配特征点对。

## (2) 去除误匹配点

在得到了匹配的特征点对后，还需要对误匹配的图像特征点对予以筛选。最为常见的去除误匹配特征点对的算法，RANSAC 算法，即随机抽样一致性检测算法。

用于图像匹配的 RANSAC 算法基本思想为：首先，随机地选取若干个匹配点对，以这些匹配点对构建匹配模型；然后，按该匹配模型计算其它匹配点对的匹配误差，当匹配误差小于给定阈值参数时，则认为这对匹配点对也满足该匹配模型，即 RANSAC 算法中的数据内点，否则为外点；最后，若在所有的匹配点对中，有超过 2/3 的匹配点对均满足该匹配模型，则可以利用所有的内点重新构造数学模型作为最后的估计模型，不满足该模型的匹配点对即为误匹配特征点，应予以排除，否则需要重新随机选择匹配点对构建数学模型，重复上述步骤。其具体过程如下：

步骤 1：获取模板图像  $a$  和采集图像  $b$  之间的所有匹配特征点对坐标向量  $(a_1, b_1), (a_2, b_2), \dots, (a_M, b_M)$ ， $a_i, b_i$  分别为图像  $a$  和图像  $b$  的第  $i$  对匹配特征点坐标， $M$  为匹配点的数量，设置计数值；

步骤 2: 随机选取  $N$  对匹配特征点对  $(a_1, b_1), (a_2, b_2), \dots, (a_N, b_N)$ , 其中  $N < M$ ;

步骤 3: 根据步骤 2 的  $N$  对匹配特征点计算两幅图像之间的转换矩阵  $H$ ;

步骤 4: 根据转换矩阵计算采集图像其余特征点的转换坐标  $\bar{b}_i = H * a_i$ ,  $i = N+1, N+2, \dots, M$ ;

步骤 5: 计算采集图像中特征点坐标和转换坐标之间的距离  $d_i = \|\bar{b}_i - b_i\|$ , 如果  $d_i < T$ , 其中  $T$  为给定阈值, 则  $k = k+1$ , 否则, 该特征点对匹配失败, 并记录。直到所有特征点距离计算完毕后执行下一步骤;

步骤 6: 如果  $k > 2M/3$ , 接受步骤 3 得到的转换矩阵, 并删除步骤 5 中记录的匹配失败的特征点对, 完成误匹配点筛选, 否则回到步骤 2, 重新选取初始匹配特征点对。

## 2) 表盘区域及指针提取

在定位到仪表之后, 为了进一步降低区域的范围, 剔除影响图像处理算法的其他区域。首先, 本系统采用颜色区域的提取, 方法仍然采用 RGB 转 HSV 空间。其次, 通过 canny 检测, 提取仪表指针。Canny 算子求边缘点的算法步骤如下:

### (1) 用高斯滤波器平滑图像

Canny 算法通常处理的图像是灰度图, 因此对于彩色图像, 必须首先进行灰度化, 以 RGB 格式的图像为例, 灰度化的公式为:

$$Gray = 0.299R + 0.587G + 0.114B$$

图像高斯滤波的实现可以用两个一维高斯核分别两次加权实现, 也就是先一维 X 方向卷积, 得到的结果再一维 Y 方向卷积, 当然也可以直接通过一个二维高斯核一次卷积实现。

### (2) 用一阶偏导有限差分计算梯度幅值和方向

令  $G$  为二维高斯函数

$$K = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

式中  $\sigma$  为标准差,  $(x, y)$  代表图像横、纵坐标点, 则沿着  $k$  方向的一阶导数导

数为

$$G_k = \frac{\partial G}{\partial k} = k \cdot \nabla G$$

对于图像  $f$ ，其边缘的法向  $k$  为

$$k = \frac{\nabla(G * f)}{|\nabla(G * f)|}$$

其中 $*$ 表示卷积。

(3) 对梯度幅值进行非极大值抑制

图像灰度值的梯度一般使用一阶有限差分来进行近似，这样就可以得图像在  $x$  和  $y$  方向上偏导数的两个矩阵。

$$P[i, j] = (f[i+1, j] - f[i, j] + f[i+1, j+1] - f[i, j+1]) / 2$$

$$Q[i, j] = (f[i, j] - f[i, j+1] + f[i+1, j] - f[i+1, j+1]) / 2$$

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2}$$

$$\theta[i, j] = \arctan(Q[i, j] / P[i, j])$$

式中  $f$  为图像灰度值， $P$  代表  $X$  方向梯度幅值， $Q$  代表  $Y$  方向梯度幅值， $M$  是该点幅值， $\theta$  是梯度方向，也就是角度。

非极大值抑制是进行边缘检测的一个重要步骤，通俗意义上是指寻找像素点局部最大值。沿着梯度方向，比较它前面和后面的梯度值进行，示意图如图 2.10 所示。

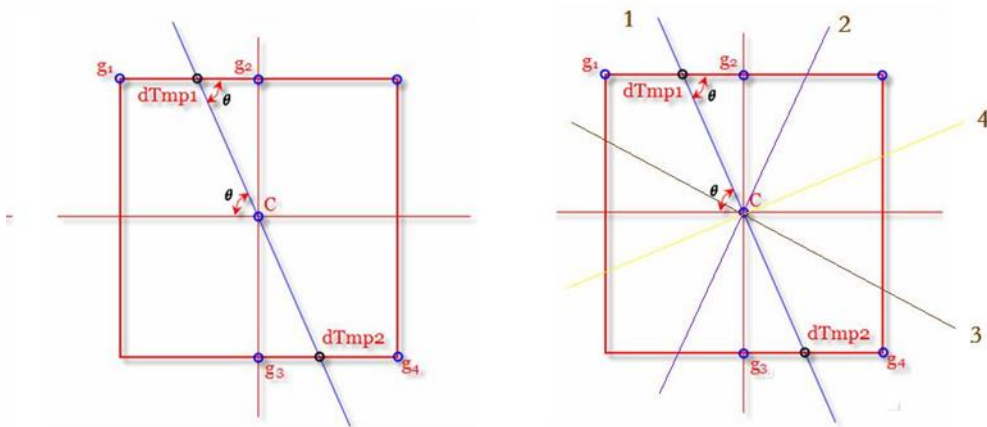


图 2.10 梯度方向寻找极大值

上图中  $g_1$ 、 $g_2$ 、 $g_3$ 、 $g_4$  都代表像素点，很明显它们是  $c$  的八邻域中的 4 个，左图中  $c$  点是我们需要判断的点，蓝色的直线是它的梯度方向，也就是说  $c$  要是

局部极大值，它的梯度幅值  $M$  需要大于直线与  $g_1g_2$  和  $g_2g_3$  的交点， $dtmp1$  和  $dtmp2$  处的梯度幅值。但是  $dtmp1$  和  $dtmp2$  不是整像素，而是亚像素，也就是坐标是浮点的，那怎么求它们的梯度幅值呢？线性插值，例如  $dtmp1$  在  $g_1$ 、 $g_2$  之间， $g_1$ 、 $g_2$  的幅值都知道，我们只要知道  $dtmp1$  在  $g_1$ 、 $g_2$  之间的比例，就能得到它的梯度幅值，而比例是可以靠夹角计算出来的，夹角又是梯度的方向。

设  $g_1$  的幅值  $M(g_1)$ ， $g_2$  的幅值  $M(g_2)$ ，则  $dtmp1$  可以很得到：

$$M(dttmp1) = w * M(g_2) + (1 - w) * M(g_1)$$

其中

$w = \text{distance}(dttmp1, g_2) / \text{distance}(g_1, g_2)$ ， $\text{distance}(g_1, g_2)$  表示两点之间的距离。实际上  $w$  是一个比例系数，这个比例系数可以通过梯度方向(幅角的正切和余切)得到。右边图中的 4 个直线就是 4 个不同的情况，情况不同， $g_1$ 、 $g_2$ 、 $g_3$ 、 $g_4$  代表  $c$  的八邻域中的 4 个坐标会有所差异，但是线性插值的原理都是一致的。

#### (4) 用双阈值算法检测和连接边缘

双阈值的选取是按照直方图来选择的，首先把梯度幅值的直方图求出来，选取占直方图总数%多少所对应的梯度幅值为高阈值，高阈值的一半为低阈值，通常最好的方式是根据实际图进行设置。

针对边缘连接，首先判断该点是否超过高阈值，然后判断该点的 8 邻域点中寻找满足超过低阈值的点，再根据此点收集新的边缘，直到整个图像边缘闭合。整个图像找完后，将非边缘点剔除，即灰度值置 0，从而完成 canny 边界检测。

### 3) 读数建模

由于巡检机器人拍摄视角的变化，容易对量程的弧度产生一定的畸变，因此，在读数建模中，本系统依然采用比例来确定最终的度数。

假定各个颜色区域的长度在数据库是已知的，并于真实的数值意义对应，即每个长度的大小值分别为

$$\text{在绿色区段: } R = \frac{h_1}{L_1} \times 1.0$$

$$\text{在黄色区段: } R = 1.0 + \frac{h_2}{L_2} \times 1.0$$

在红色区域:  $R = 2.0 + \frac{h_3}{L_3} \times 1.0$

其中,  $h_1, h_2, h_3$  分别是各个区域的指针位置, 例如图 2.11 中  $h_1$ ;  $L_1, L_2, L_3$  分别是颜色区段长度。

基于此, 当检测到  $L_i'$  和  $h_i'$ , 则可计算读数为:

在绿色区段:  $R = \frac{h_1'}{L_1'} \times 1.0$

在黄色区段:  $R = 1.0 + \frac{h_2'}{L_2'} \times 1.0$

在红色区域:  $R = 2.0 + \frac{h_3'}{L_3'} \times 1.0$

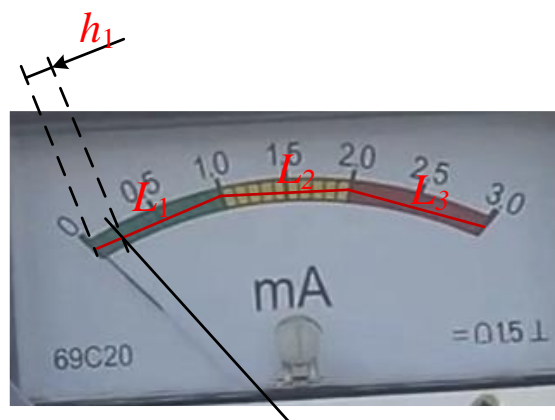
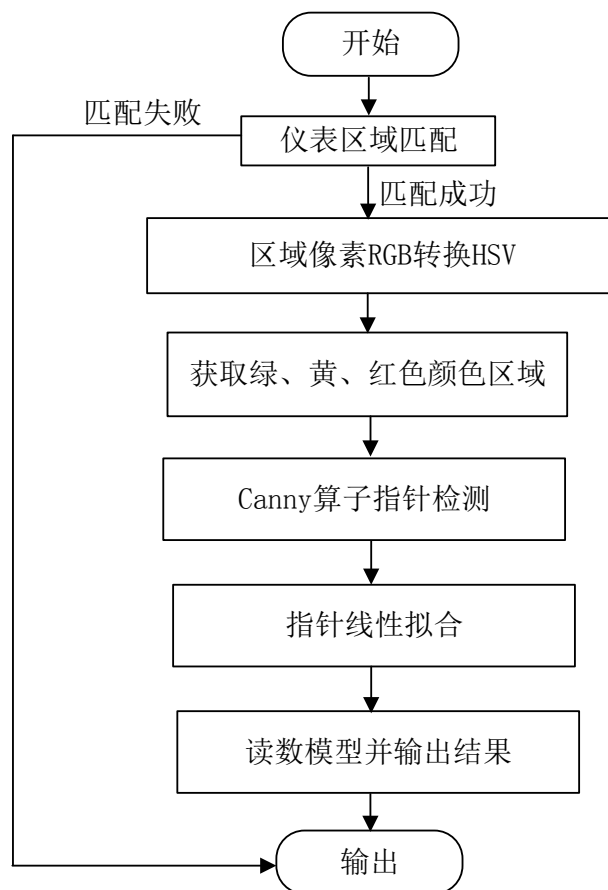


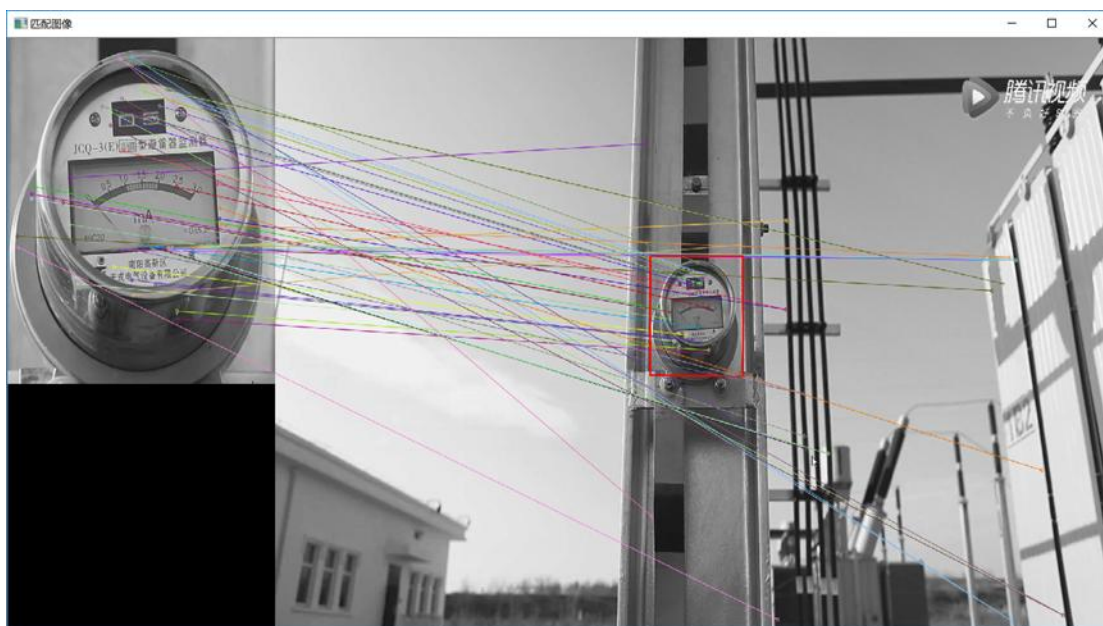
图 2.11 表盘读数建模

#### 4) 算法流程

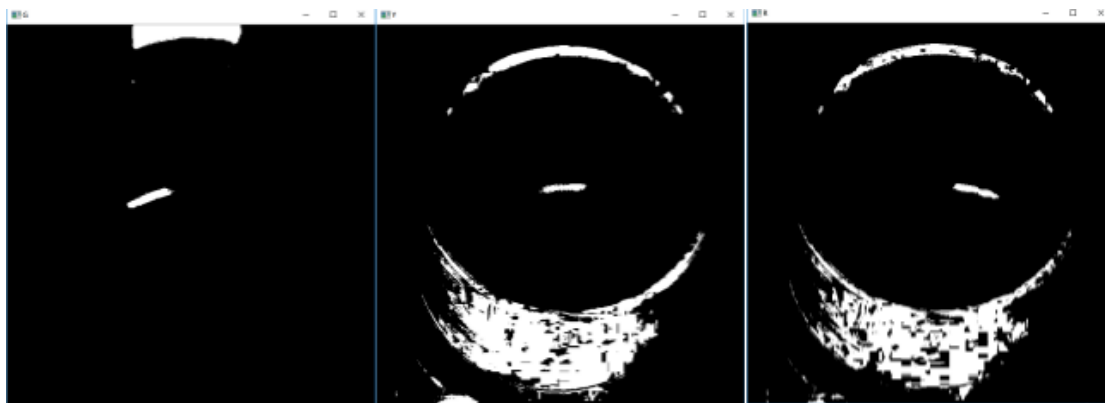


## 5) 实测结果

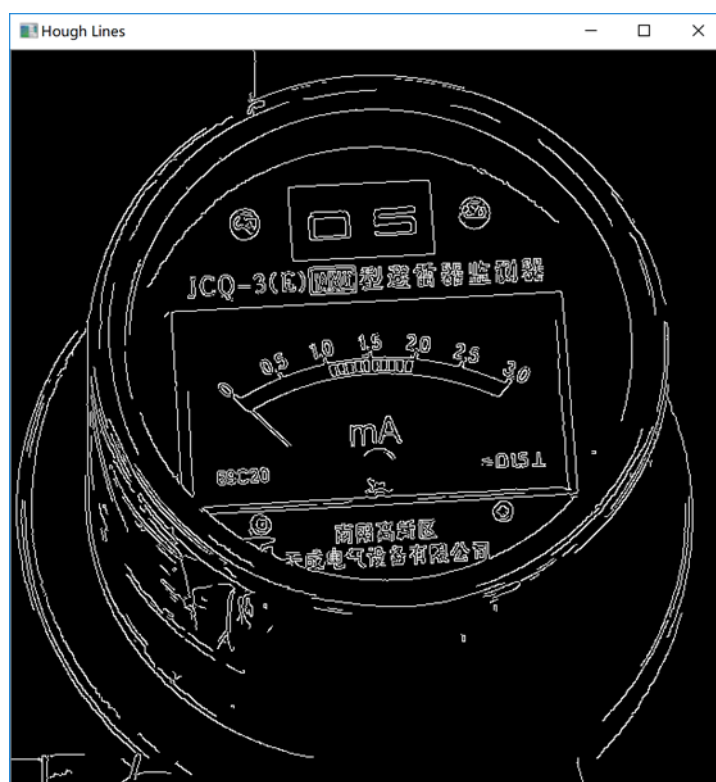
### 第一步，仪表区域匹配



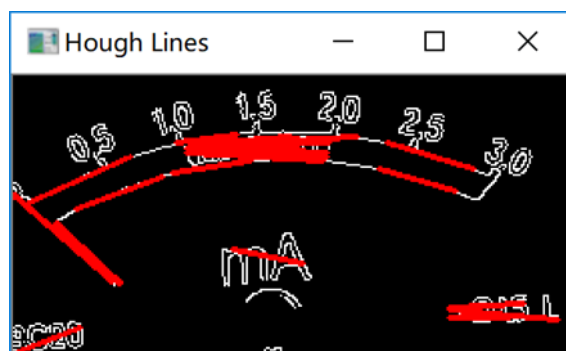
### 第二步，监视器仪表指示区域提取



第三步，Canny 区域边界检测



第四步，指针拟合



第五步，通过指针与区域的相关性，计算读数



### 2.2.3 断路器仪表

下图 2.12 是一种圆盘指针式仪表。这种仪表具有圆的特性。当设计仪表拍摄偏移等问题时,可以采用仿射变换或透视变换的方式来对图像进行纠正。另外,在对图像进行圆定位,指针定位的过程中,需要进行图像的灰度化,平滑去噪,甚至形态学腐蚀膨胀等操作,便于提高读数准确度。



图 2.12 圆盘指针式仪表

#### 1) 仪表图像的预处理及匹配

为了能够检测到仪表,通常按先前提到的 Surf 算子等特征点匹配的方式,从图像中寻找仪表区域。

由于表盘的圆形特性,在图像处理中,可以通过检测圆的方式确定表盘,从而为后续指针的直线检测,角度的准确读取提供信息。本系统采用比较成熟的 Hough 圆以及直线检测算法。

Hough 检测的基本思想是将图像从原图像空间变换到参数空间,在参数空间中,使用大多数边界点都满足的某种参数形式作为图像中的曲线的描述,它通过



设置累加器对参数进行累积，其峰值对应的点就是所需要的信息。

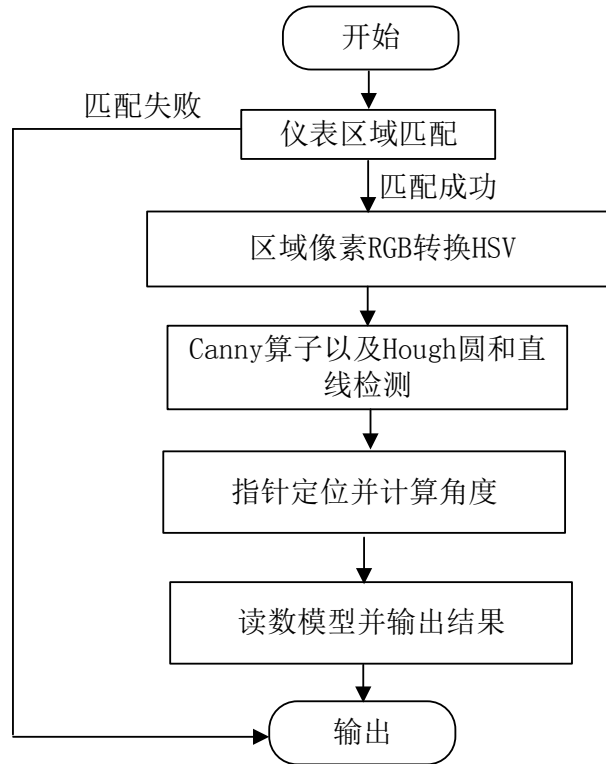
## 2) 表盘指针角度计算及数据读取

由于整个表指针范围分布并不是 360 度分布，而得到的圆通常是 360 读进行绘制，因此对于数据的计算，需要在标准库中建立初始为 0 所对用的角  $\alpha_0$ 。当检测到的角度为  $\alpha'$  时，则读数结果为

$$V = L * \frac{\alpha' - \alpha_0}{\alpha_L - \alpha_0}$$

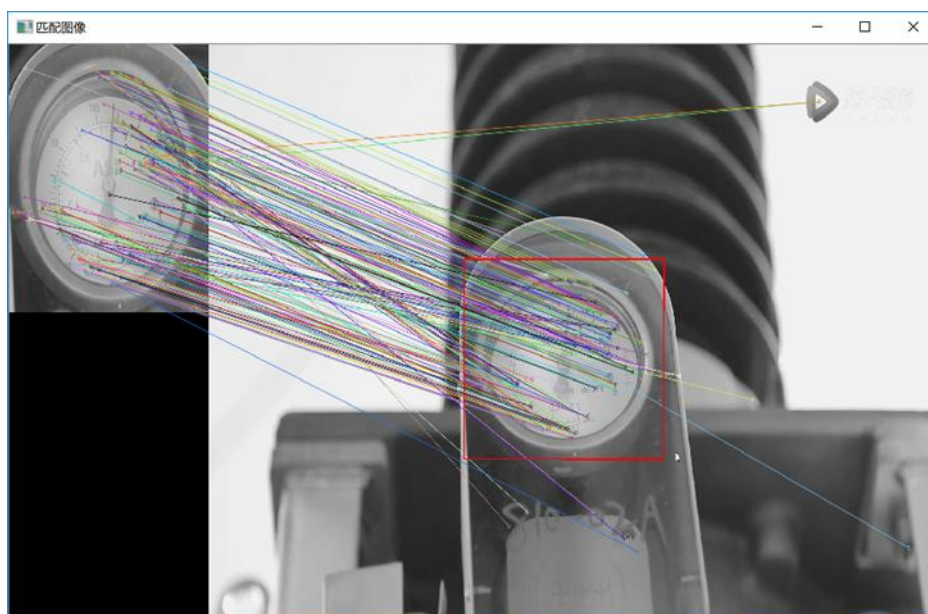
时钟  $L$  是整个仪表量程， $\alpha_L$  是最大量程对于的角度。

## 3) 算法流程

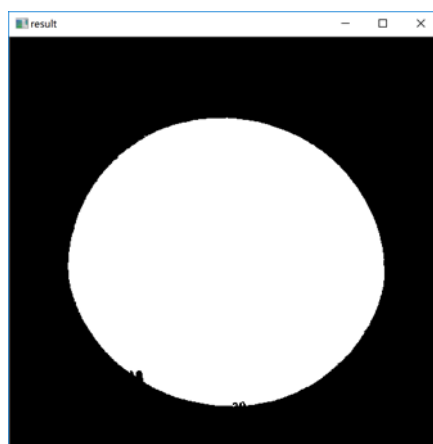


## 4) 测试结果

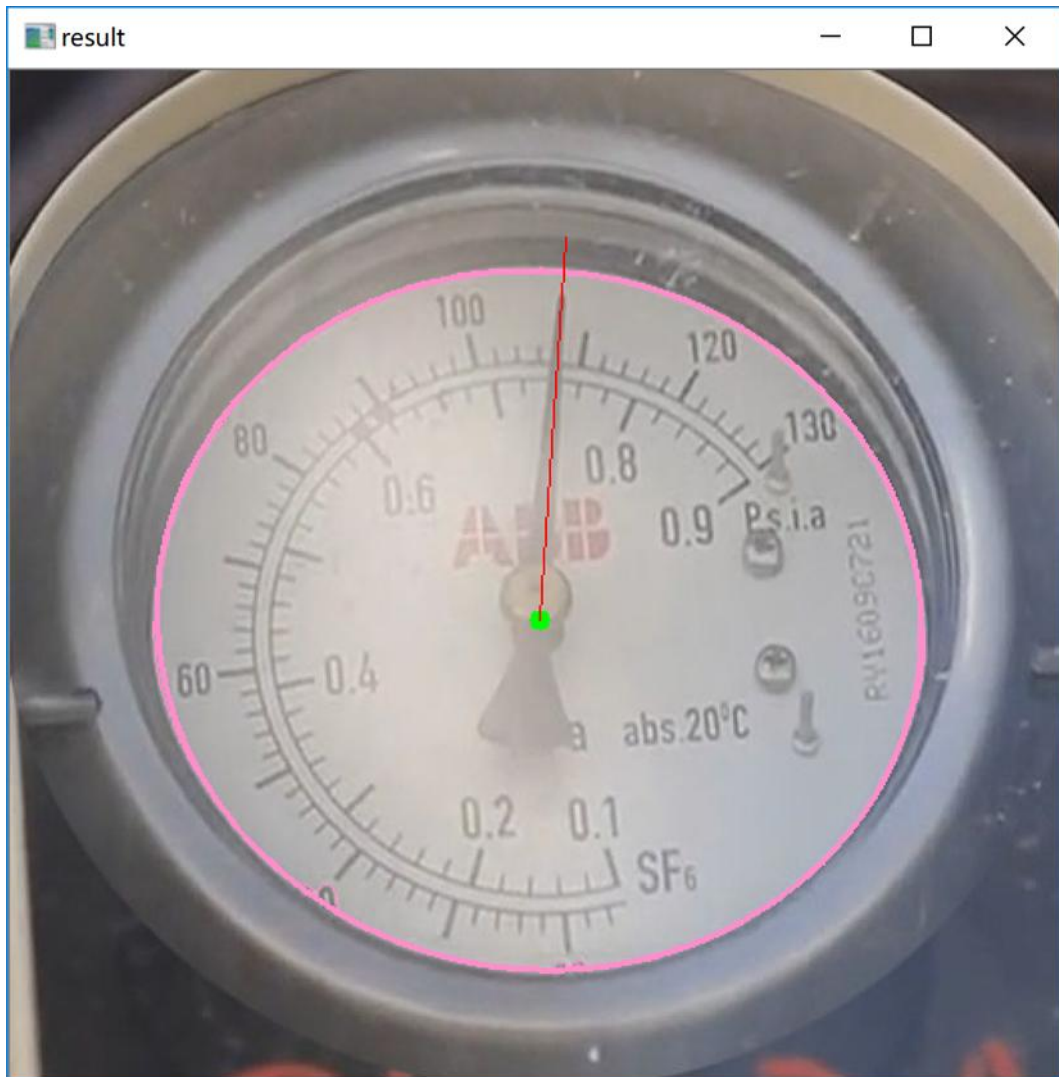
第一步：目标匹配



第二步：表盘区域提取



第三步：表针提取



第四步：读数结果  $\alpha'=84^\circ$ ，最终得到的结果是  $L=108.8322$

## 2.3 红外测温

### 2.3.1 绝缘子测温

由于红外图像只呈现温度场信息，对于具体需要对哪种设备或装置进行温度检测，直接采用红外图像处理方式难度角度。因此，通过需要利用可见光图像处理，先寻找和定位目标体，然后通过红外图像分割的方式，提取区域的温度场信息。

#### 1) 绝缘子匹配及红外图像处理

图 3.1 为待检测的目标体——绝缘子。因 Surf 特征点检测的效率相对较高，故也采用该算法作为匹配算法。

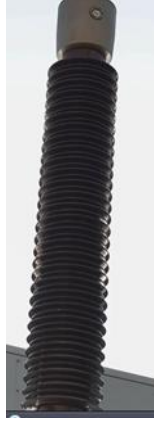


图 3.1 绝缘子

关于 Surf 特征点的检测原理，这里不做赘述。待匹配到目标体之后，红外成像仪开始工作，并对目标体区域进行分割，准确提取绝缘子温度信息。本系统采用扩展的 Meanshift 聚类分割。

Mean shift 算法通常被称为均值漂移算法，在聚类、图像平滑、分割、跟踪等方面有着广泛的应用。本质上，它可归结为一种迭代搜索算法，即先根据数据集计算出当前点的均值偏移，然后以此为新的起始点，继续移动，直至满足收敛条件。因此，在遍历样本过程中，每个类中的样本点都将收敛至其概率密度极大值处，并可将样本类别进行自适应划分。

为了便于描述，假设采样点  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $i=1, \dots, n$ ，则点  $\mathbf{x}$  处的概率密度估计如下式所示

$$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{\mathbf{x}_i \in S_h} K\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)$$

式中  $S_h$  是一个固定带宽为  $h$  的高维球区域

$$S_h(\mathbf{x}) = \{\mathbf{y} | \|\mathbf{y} - \mathbf{x}\|^2 \leq h^2\}$$

$d$  表示维数， $K(\cdot)$  为径向对称核函数

$$K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2)$$

$c_{k,d}$  为归一化常数； $k(\mathbf{x})$  为核函数  $K(\mathbf{x})$  的剖面函数。

样本点  $\mathbf{x}_i$  对当前中心点  $\mathbf{x}$  的贡献与核函数密切相关，考虑到在 Mean shift 中所常用的两类核函数(a)单位均匀核函数：

$$K(\mathbf{x}) = \begin{cases} 1 & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{if } \|\mathbf{x}\| \geq 1 \end{cases}$$

和(b)单位高斯核函数:

$$K(\mathbf{x}) = e^{-\|\mathbf{x}\|^2}$$

单位均匀核使得各个样本点到中心点的贡献值一样,不利于体现各个样本点对概率密度估计的贡献,因此一般采用高斯核作为 Mean shift 核函数。

为了寻找极大密度值点,通常令

$$\nabla \mu(\mathbf{x}) = 0$$

可得到

$$\nabla \mu(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \left[ \sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \right] \times m_h(\mathbf{x})$$

式中  $g(x) = -k'(x)$ ,  $m_h(\mathbf{x})$  为均值漂移向量

$$m_h(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}$$

从中可以发现,均值漂移向量总是指向密度增长最快的方向。因此,当给定初始点, Mean shift 就会根据下面两个步骤逐次迭代

step1: Mean shift 向量计算, 改变当前中心位置

$$\mathbf{x}^{t+1} = \mathbf{x}^t + m_h(\mathbf{x})$$

step2: 平移窗口, 重新计算概率密度;

最终收敛到概率密度极大值处, 即满足  $\|\mathbf{x}^{t+1} - \mathbf{x}^t\| < \varepsilon$ 。

为了清晰的描述其迭代过程, 其中心点漂移如图 3.2 所示。

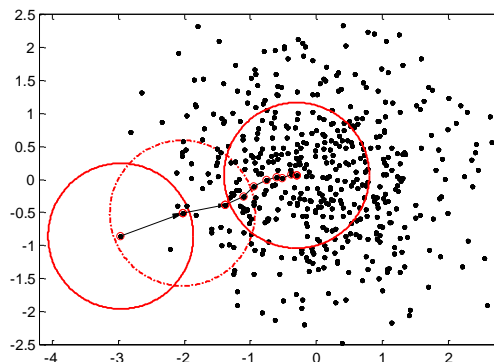


图 3.2 聚类方法中的中心点迭代过程

从上式可以看出，只要是落入  $S_h$  的样本点，无论其离当前中心点  $\mathbf{x}$  远近，对最终的  $M_h(\mathbf{x})$  计算的贡献值与核函数定义有关。一般而言，离  $\mathbf{x}$  越近的采样点对估计  $\mathbf{x}$  周围的统计特性越有效。另外，考虑到每个样本点都可赋一定权值或比重，使得最终的聚类效果更优。显然，如果将每个样本点权值都设置相同，便成为了传统 Mean shift。为此，本文方法将从以下两个方面进行改进。

#### (1) 带权重的均值漂移

对于一个概率密度函数  $f(\mathbf{x})$ ，已知  $d$  维空间中  $n$  个点  $\mathbf{x}_i$ ， $i=1, 2, \dots, n$ ，考虑样本点权值，概率密度估计可重写为

$$\mu(\mathbf{x}) = \frac{\sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \omega(\mathbf{x}_i)}{h^d \sum_{i=1}^n \omega(\mathbf{x}_i)}$$

$\omega(\mathbf{x}_i) \geq 0$  是赋给每一个样本的权重，则概率密度函数的梯度估计为

$$\begin{aligned} \nabla \mu(\mathbf{x}) &= \frac{2}{h^2} \left[ \frac{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}_i - \mathbf{x}}{h}\right\|\right) \omega(\mathbf{x}_i)}{h^d \sum_{i=1}^n \omega(\mathbf{x}_i)} \right] m_h(\mathbf{x}) \\ m_h(\mathbf{x}) &= \left[ \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}_i - \mathbf{x}}{h}\right\|\right) \omega(\mathbf{x}_i)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}_i - \mathbf{x}}{h}\right\|\right) \omega(\mathbf{x}_i)} - \mathbf{x} \right] \end{aligned}$$

相比于前者所示的均值漂移向量，上式所示的均值漂移向量会因权重的变化，使得当前聚类中心朝着权重密度极大以及概率密度极大方向漂移。其中针对像素点权重  $\omega$  的设置，本文根据像素点  $\mathbf{x}_i$  的邻域方差进行设置

$$\omega(\mathbf{x}_i) = c_r / \sum_{\mathbf{y} \in N_i} \|\mathbf{x}_i^r - \mathbf{y}^r\|^2$$

式中  $c_r$  为归一化常量， $N_i$  为像素  $\mathbf{x}_i$  的邻域。显然当邻域特征  $\mathbf{x}_r$  与对应中心点  $\mathbf{x}_i$  相近，则权重  $\omega(\mathbf{x}_i)$  越大；而当邻域特征  $\mathbf{x}_r$  与中心点  $\mathbf{x}_i$  相比差异较大，则  $\omega(\mathbf{x}_i)$  较小，从而使得均值朝方差变化较小的方向漂移。

#### (2) 自高而低的分割机制

图像通常可以由二维网格上坐标和对应像素值组合而成，因此像素点信息可以表示为向量  $\mathbf{x} = (\mathbf{x}_s, \mathbf{x}_r)$ ，其中  $\mathbf{x}_s$  表示网格点的坐标， $\mathbf{x}_r$  表示该网格点上颜色

特征。因红外图像为热图像，只显示亮度信息。

为了能够高效提取相似区域，本文在采用均值漂移进行图像内相似像素聚类时，首先需要构建联合灰度和空间特征的核函数；其次，将迭代的方式由迭代收敛某个聚类中心点转换为收敛于区域内任意点的方式，其中区域的提取融合了自高向低的分割机制。其具体如下：

规定若像素点  $x_i$  的  $x_r$  特征与对应像素点  $x$  的  $x_r$  特征相近，且  $x_i$  的位置距离像素点  $x$  越近，概率密度则越高，因此采用空间和特征的核函数的组合来定义一个新的核函数  $K_{h_r, h_s}$

$$K_{h_s, h_r} = \frac{C}{h_s^2 h_r^p} k\left(\|x^s/h_s\|^2\right) k\left(\|x^r/h_r\|^2\right)$$

其中  $h_s$  为图像空间带宽， $h_r$  是灰度带宽， $C$  是一个归一化常数。

规定阈值设置为均值偏移聚类得到的区域均值，然而利用这个阈值，对图像进行全局阈值分割处理，以确保当前聚类的区域为图像的局部区域，从而大大减少 Mean shift 搜寻区域聚类中心的迭代次数，同时也可以有效地获取图像中其他的相似区域。因红外电气设备故障图像中，高亮区域通常代表有故障的发生，因此，采用阈值分割机制，可以直接将图像分割成二值图像，其中 1 表示故障发生区域，0 表示其他区域，然后计算故障发生区域均值，以此通过扩展的均值漂移算法对邻域相似像素进行聚类。

整个实现过程如下：

step(1)：采用均值滤波对原始图像进行平滑，去除噪声，同时计算各个像素点权值  $\omega(x_i)$ ；

step(2)：赋初始阈值  $T$  为红外图像中最高灰度值，以此作为本方法初始聚类区域，并标记；

step(3)：计算已经聚类的区域之外满足图像空间带宽  $h_s$  内未聚类的像素点  $x_i$  的密度估计，针对其邻域进行聚类；

step(4)：将聚类区域的灰度均值赋给阈值  $T$ ，并对图像进行划分，获取图像空间中其他可能出现的高亮区域；同时保留已经标记的像素点；

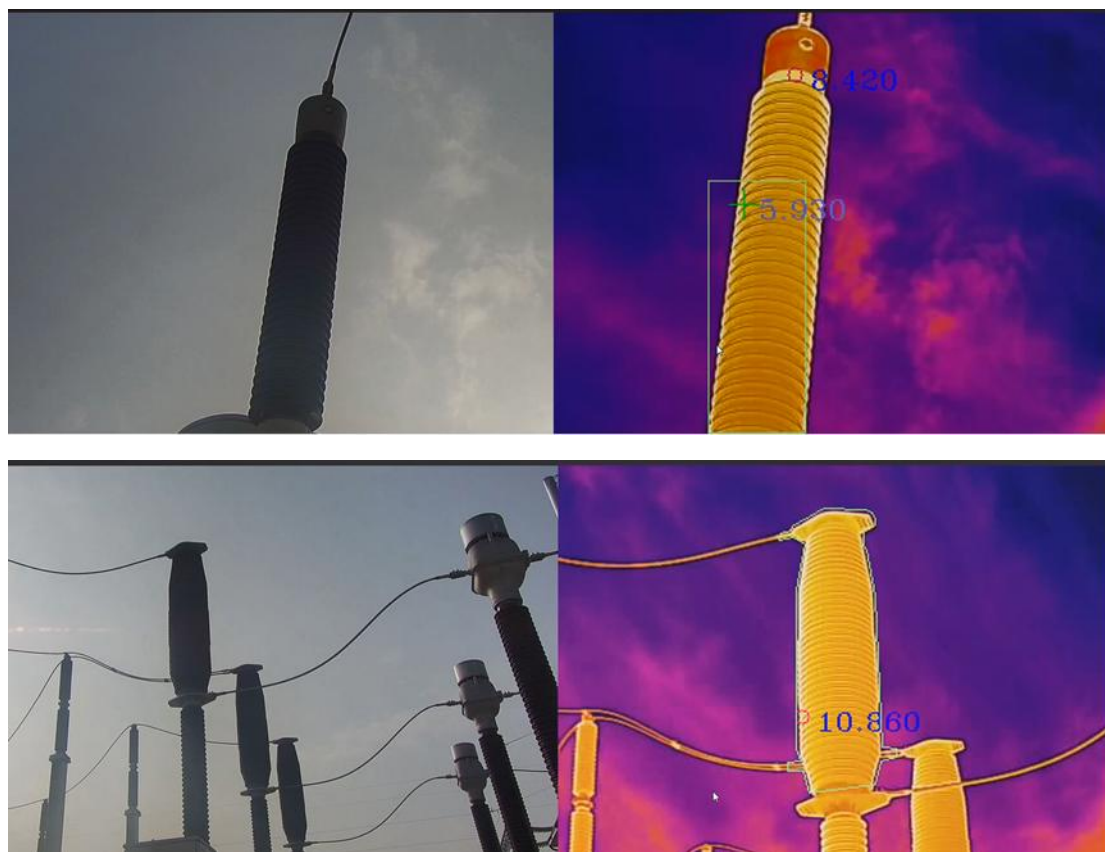
step(5)：如果整个聚类区域不再发生变化，则停止迭代，输出故障区域结果，否则转向 step(3)，继续循环。

## 2) 红外图像温度转换

在红外图像中，灰度值的大小与温度是一一对应关系，因此可直接通过灰度值转换关系，获取温度值。

为了得到图像灰度值和物体温度值之间的对应关系，需要做黑体标定试验。通过实验的方式确定黑体灰度值  $U$  与  $T$  的具体关系，即所谓的黑体温度——灰度转换曲线。然后，通过线性拟合的方式，确定曲线关系。

### 2.3.2 测试结果



## 三、导航系统软件

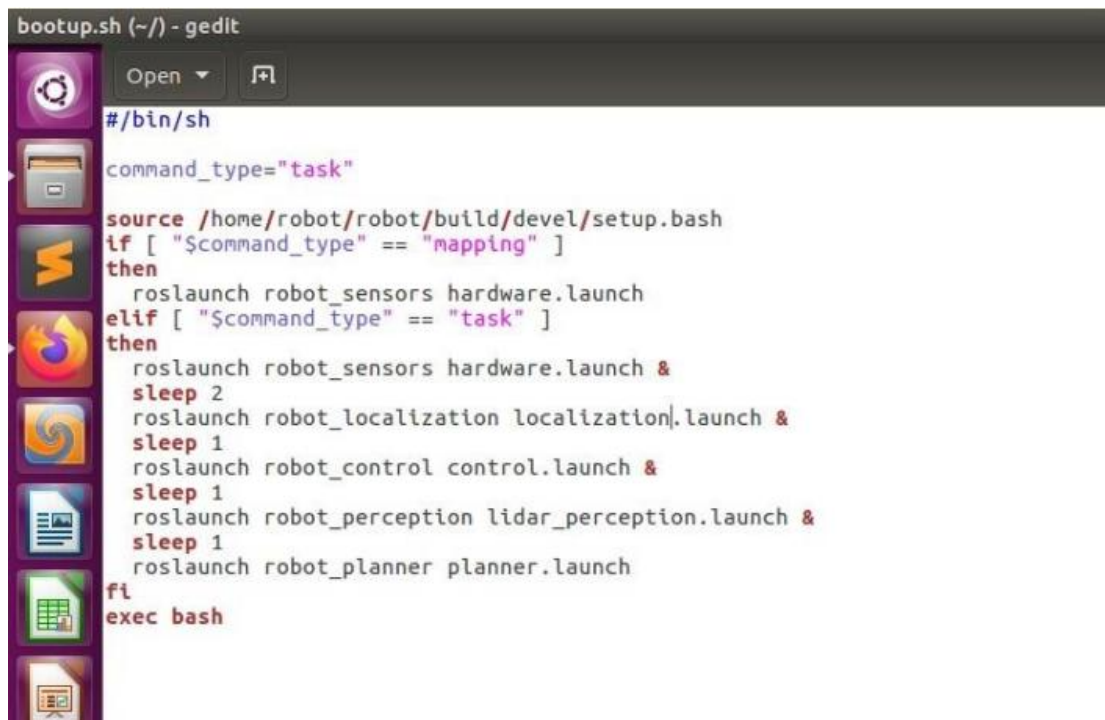
### 3.1 使用环境与配置

#### 3.1.1 开机启动配置

当导航计算机被用于建图或执行任务时，需要对开机启动内容进行配置。

打开 home 下的 `bootup.sh`:





```
bootup.sh (~/) - gedit
#/bin/sh
command_type="task"
source /home/robot/robot/build/devel/setup.bash
if [ "$command_type" == "mapping" ]
then
    roslaunch robot_sensors hardware.launch
elif [ "$command_type" == "task" ]
then
    roslaunch robot_sensors hardware.launch &
    sleep 2
    roslaunch robot_localization localization.launch &
    sleep 1
    roslaunch robot_control control.launch &
    sleep 1
    roslaunch robot_perception lidar_perception.launch &
    sleep 1
    roslaunch robot_planner planner.launch
fi
exec bash
```

图 3.1 配置开机启动

如果需要建图，则修改 `command_type=" mapping"`，如果需要执行任务，则 `command_type=" task"`，均为小写。配置完成后重启即可使用。

### 3.1.2 配置上位机

配置上位机软件电脑的方法（包括配置主从机）：

1) 将使用者的笔记本用网线接入路由器，也可以接无线路由，无线网络 User name: root，密码 123456789，如果要建图采数据由于数据量较大，请使用有线。

2) 连接以太网端口到使用者的 pc，设置 ip:192.168.1.xxx（根据需要自行设置，不能设置为 102,00,222）。

3) 进行建图和导航任务时候的开机启动配置方式。

4) 在终端输入：ping 192.168.1.102，如果 ping 通说明设置成功。

5) 打开终端，输入：sudo gedit ~/.bashrc，并在末尾加入以下内容：

```
export ROS_MASTER_URI=http://192.168.1.102:11311
```

```
export ROS_IP=192.168.1.1xxx（此处为本机 IP）
```

```
source /opt/ros/kinetic/setup.bash
```

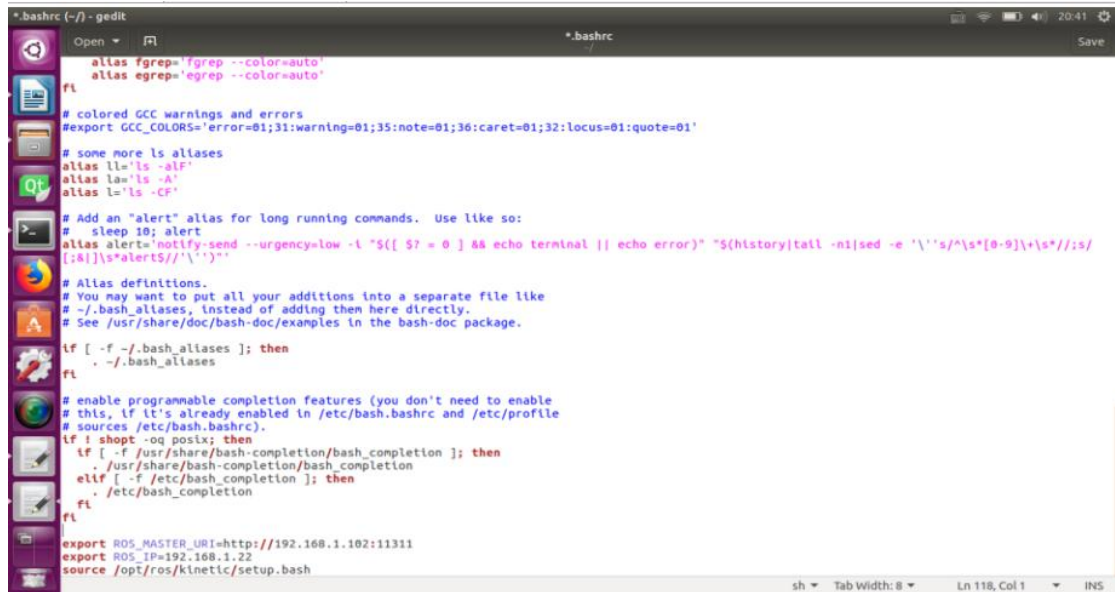


图 3.2 配置上位机

6) 在 terminal 中输入 `sudo gedit /etc/hosts`, 加入 `192.168.1.102robot-PC`, 并保存。

7) 保存，完成配置。

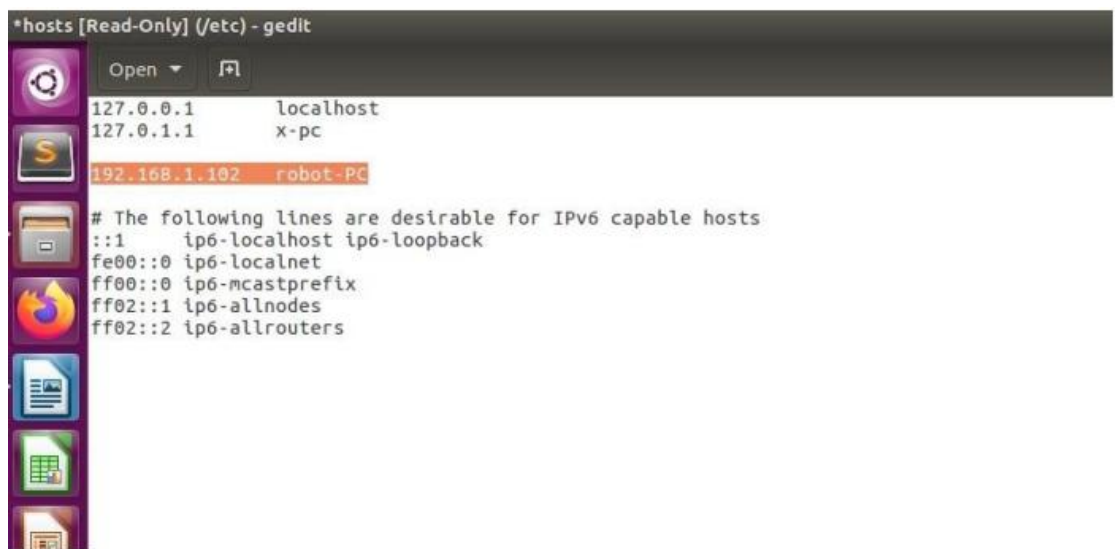


图 3.3 配置上位机

### 3.2 在线建图步骤

建图时候需要注意以下几点：建图过程不能中断；环境发生改变 30%需要重新建图，否则定位精度下降甚至定位丢失；建图初始位置选择很重要，建图时机器人启动的位置为地图原点，机器人启动的朝向为地图 x 轴朝向；当机器人首次使用，在关机状态被人为移动，或定位丢失后，需要将机器人置于原点重置并初

始化，因此建图时机器人启动的位置必须为容易被记住和初始化的位置；对于一个有很多岔路的场景，应该选取方便机器人进行回环检测的路线。建图具体操作步骤如下：

### 1) 在线建图，通过 rviz 观察实时 2D 地图、匹配点云和约束关系

选择建图开机启动方式为 `command_type="mapping"`，重启电脑，将机器人放置于一个能被初始化的位置，并发送 `start_mapping` 指令，如果返回了 `start_mapping_succeed`，则说明建图任务正常运行，此时通过遥控器控制机器人遍历机器人执行任务时需要经过的区域，即可生成需要的地图。整个过程可以通过 `ros rviz` 观察 2D 地图、匹配点云、约束关系等信息。运行 `roslaunch robot_map mapping_visualization.launch`，即可看到如下画面，其中灰底色的的是 2D 地图，彩色点云是匹配点云，彩色线段为约束关系。

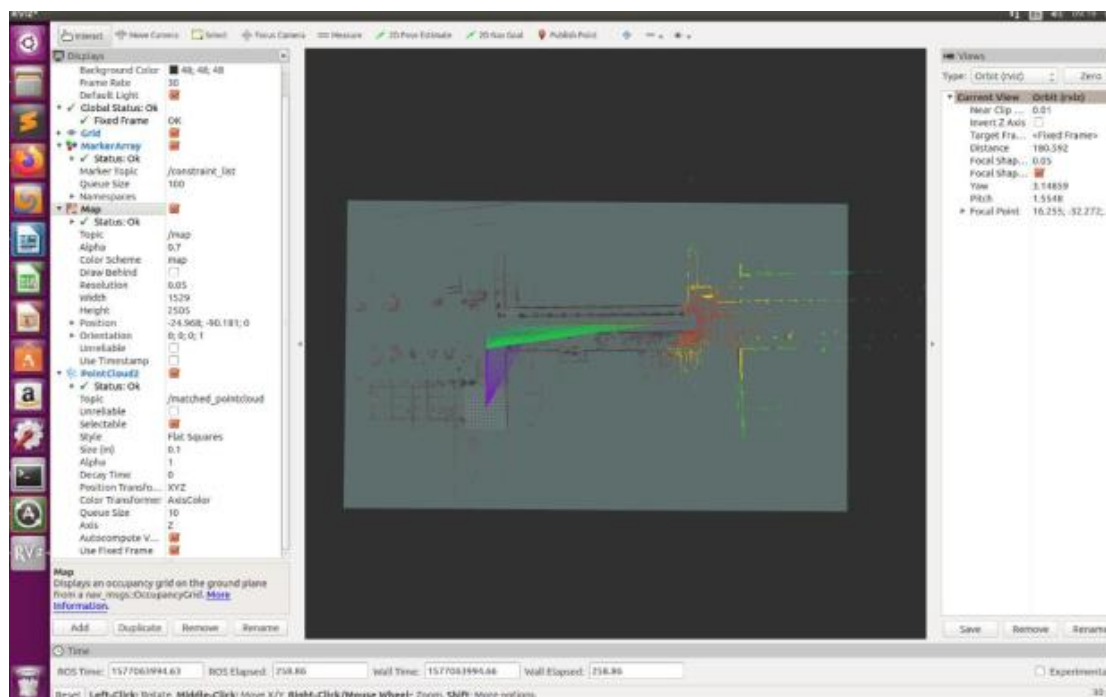


图 3.4 在线建图

### 2) 建图完成后，生成 2D 地图和配置文件

建图完成后，将机器人停止，并发送 `finish_mapping` 指令，如果返回 `finish_mapping_succeed`，则说明建图任务正常结束，此时，等待上位机软件依次收到以下调试信息：

Finish generate map files!

Received path!

Finish point cloud sampling!

Finish build 2D map!

说明所有需要的文件都已生成（需要等待一段时间），此时，建图流程完成。以上调试信息，也可使用 `rostopic echo /debug_info` 进行查看。

### 3) 如何判断建图成功

地图无重影，尤其在机器人重复经过某个区域时，机器人能够实现回环检测，则说明建图成功，否则需要重新建图。

## 3.3 地图标注步骤

标注工具主要功能分为四个部分：基本显示功能，生成新的导航地图，标注导航地图，生成新的任务和导航点文件。操作界面如下图所示：

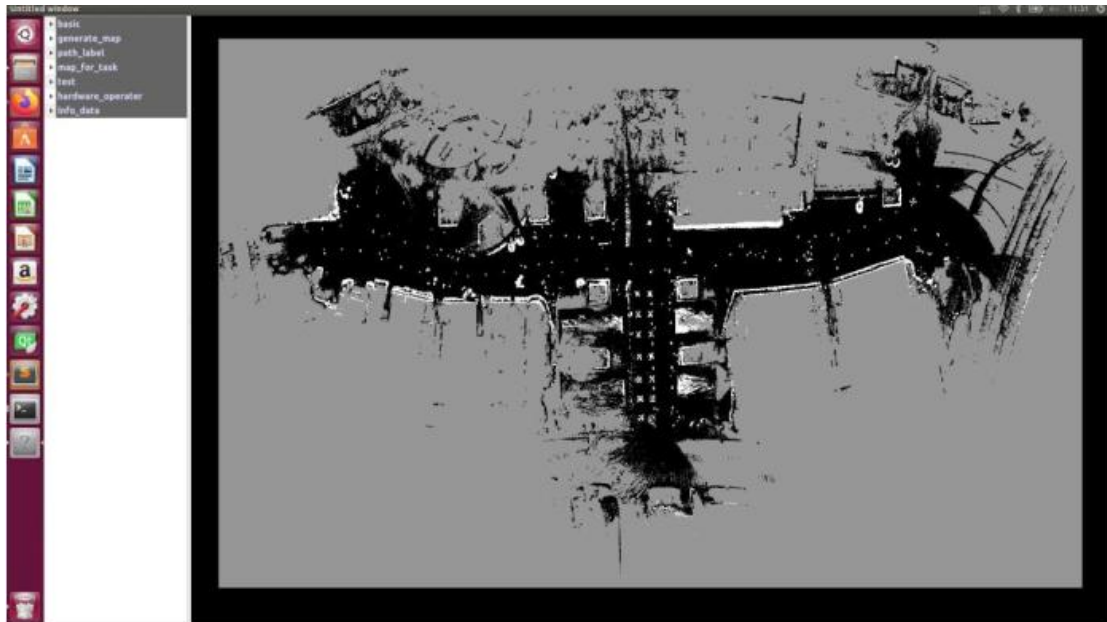
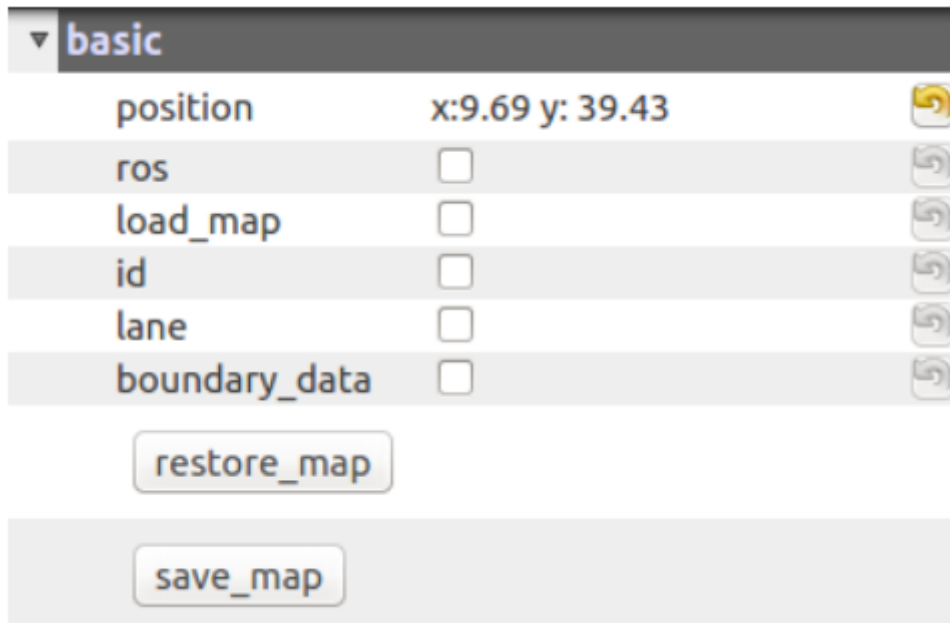


图 3.5 地图标注操作界面

### 3.3.1 基本显示功能



position: 显示鼠标所在地形对于地图的实时坐标;

ros: 是否显示机器人的坐标

load\_map: 加载新的二维地图图片;

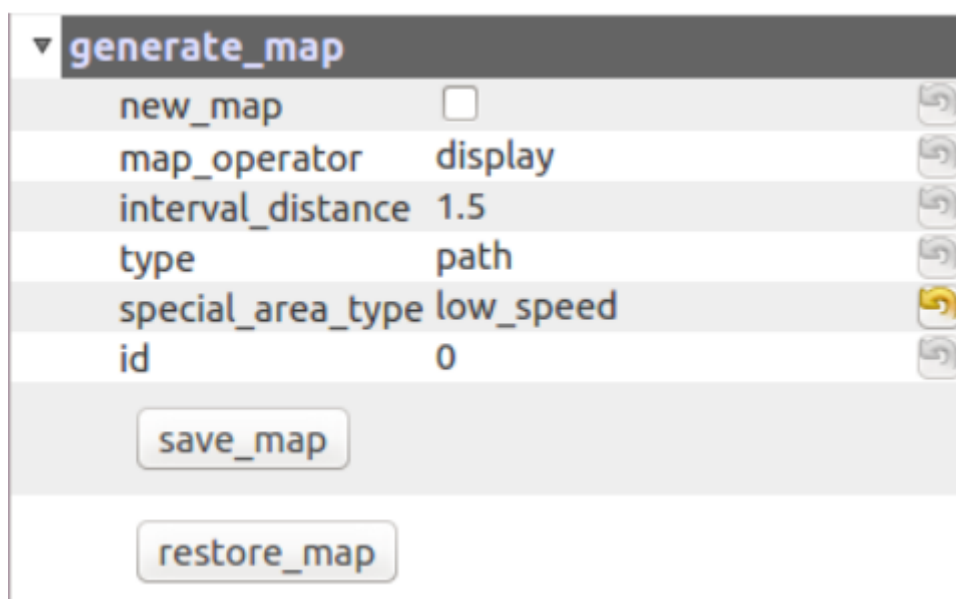
id: 显示已标记好的导航地图各元素的 id;

lane: 显示已有导航地图中的各元素;

restore\_map: 加载新的已有导航地图;

save\_map: 存储地图。

### 3.3.2 生成新的导航地图



new\_map: 选择新建导航地图;

map\_operator: 对于新建地图的各种操作, 包含显示, 增加, 删除, 修改各元素以及清空显示;

interval\_distance: 用于设置道路平滑的间隔;

type: 元素类型, 包括道路, 不可行驶区域以及特殊区域;

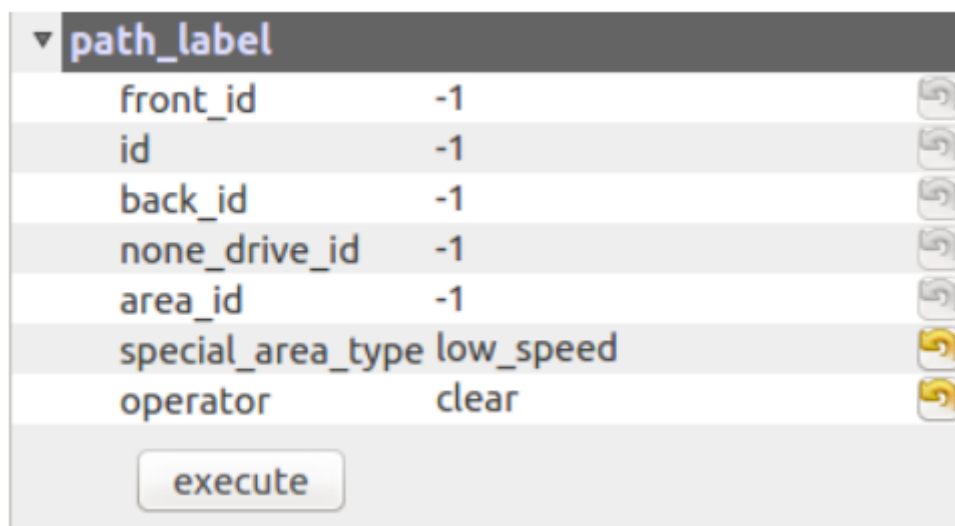
special\_area\_type: 特殊区域的类型, 包括减速以及特殊感知区域;

id: 索引并显示各元素;

save\_map: 保存新建地图;

restore\_map: 加载已有地图。

### 3.3.3 标注导航地图



当生成一个新的导航地图后还需要调整各条道以及道路之间的连接处, 使道路之间的连接平滑。

front\_id, id, back\_id: 功能相同, 索引导航地图道路并显示, 同时显示三条道路, 便于调整和修改, 具体操作方式见相关视频;

none\_drive\_id: 索引不可行驶区域并调整;

area\_id: 索引特殊区域并调整;

special\_area\_type: 显示特殊区域类型;

operator: 各种操作方式, 包括清除显示, 增加删除道路, 不可行驶区域, 特殊区域, 重新生成道路之间的关系, 以及加载和保存地图;

execute: 当完成选择类型后, 点击此按钮执行相应功能

### 3.3.4 添加任务和导航点

The screenshot shows a configuration window for a task. It has a dark header bar with a dropdown menu set to 'map\_for\_task'. Below this, there are several input fields and buttons. The 'key\_points' section is expanded, showing fields for speed\_limit (1.5), stop\_time (10), point\_angle (0), name, point\_id (0), point\_action (pose), and point (clear). There are buttons for 'set\_angle' and 'execute' below these fields. The 'nav\_points' section is also expanded, showing an 'action' field set to 'clear' and an 'execute' button below it.

| map_for_task |       |
|--------------|-------|
| map_name     |       |
| map_id       | 0     |
| ▼ key_points |       |
| speed_limit  | 1.5   |
| stop_time    | 10    |
| point_angle  | 0     |
| name         |       |
| point_id     | 0     |
| point_action | pose  |
| point        | clear |
| set_angle    |       |
| execute      |       |
| ▼ nav_points |       |
| action       | clear |
| execute      |       |

map\_name: 设置任务地图名称

map\_id: 设置任务地图 id;

key\_points: 任务点设置

speed\_limit: 设置最高限速;

stop\_time: 设置等待时间;

point\_angle: 设置任务点角度;

name: 设置任务点名称;

point\_id: 设置任务点 id;

point\_action: 设置任务类型;

point: 对任务点的操作, 包括清除, 显示, 增加, 修改, 删除, 加载和保存任务地图;

set\_angle: 获取机器人的实时角度;

execute: 执行对任务点的操作;

nav\_points: 导航点设置

action: 包括清除显示, 增加导航点, 显示导航点, 保存所有导航点文件, 加载已有导航点文件;

execute: 针对选择的动作进行执行。

## 3.4 执行任务操作步骤

### 3.4.1 基本显示功能

同 3.3 节。

### 3.4.2 任务显示



load\_task\_map: 加载已有任务地图;

task\_map\_id: 索引任务地图并显示索引点;

load\_nav\_point: 加载导航任务文件;

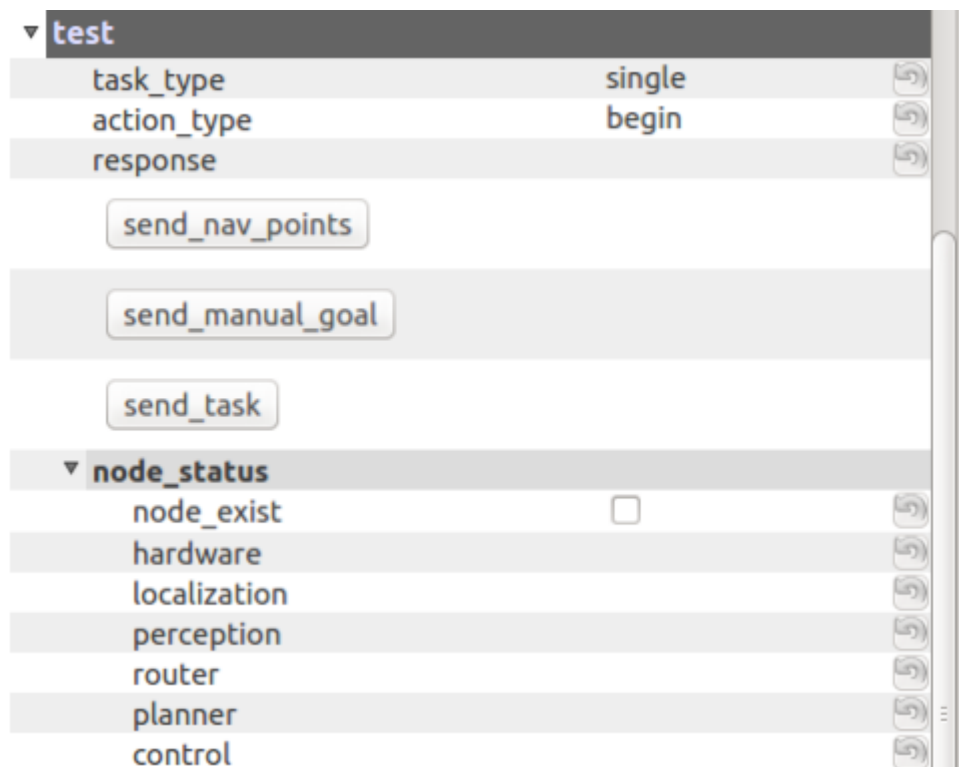
nav\_point\_id: 导航点索引并显示;

nav\_type: 显示导航点动作类型;

clear\_display: 清除所有导航点显示。

### 3.4.3 发送任务点





task\_type: 任务执行类型，分为单次和循环；

action\_type: 任务操作类型，包括开始，暂停，继续，取消以及完成；

response: 发送任务机器操作命令之后下位机反馈内容显示；

send\_nav\_points: 发送任务；

send\_manual\_goal: 发送手动点；

send\_task: 发送操作命令；

node\_status: 下位机状态反馈

node\_exist: 发送命令，测试下位机节点连接；

hardware: 硬件节点连接成功；

localization: 定位节点连接成功；

perception: 感知节点连接成功；

router: 路由节点连接成功；

planner: 规划节点连接成功；

control: 控制节点连接成功。

### 3.4.4 硬件操作

▼ hardware\_operater

auto/stop

☐

operator

none

response

generate\_map\_info

execute

auto/stop: 自动/软急停;

operator: 包括开始, 完成建图, 重置, 关机, 重启, 以及清除等操作;

response: 执行操作的反馈显示;

generate\_map\_info: 建地图反馈的信息;

execute: 执行 operator 选择的操作。

### 3.4.5 任务信息显示

▼ info\_data

▼ task\_info

task\_status

invalid

remain\_point

0

▼ robot\_info

robot\_angle

0

robot\_speed

0

front\_wheel\_angle

0

robot\_state

☐

control\_type

battery\_voltage

0

▼ fault\_info

can\_communication\_fault

☐

front\_wheel\_encoder\_fault

☐

remote\_control\_lost\_fault

☐

battery\_underload\_fault

☐

battery\_overload\_fault

☐

first\_motor\_fault

☐

second\_motor\_fault

☐

third\_motor\_fault

☐

forth\_motor\_fault

☐

driver\_over\_temperature\_fault

☐

driver\_over\_current\_fault

☐

▼ frequency

lidar

0

lidar\_points

0

imu

0

gnss

0

odometry

0

ultrasonic

0

task\_info

task\_status:任务状态;

remain\_point: 剩余任务点;

robot\_info

robot\_angle: 机器人角度;

robot\_speed: 机器人速度;

front\_wheel\_angle: 机器人前轮角度;

robot\_state:机器人状态;

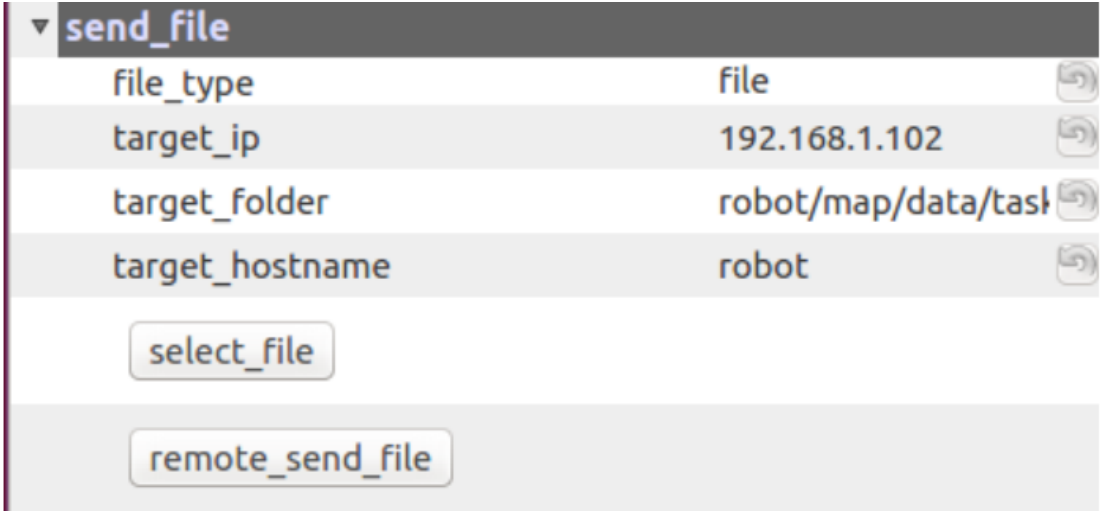
control\_type: 控制类型;

battery\_voltage: 电池电压;

fault\_info: 底盘反馈的错误信息;

frequency: 传感器发布的频率信息。

### 3.4.6 远程发送文件



| send_file       |                     |
|-----------------|---------------------|
| file_type       | file                |
| target_ip       | 192.168.1.102       |
| target_folder   | robot/map/data/task |
| target_hostname | robot               |

select\_file

remote\_send\_file

file\_type: 文件类型, 包括单个, 所有文件以及文件夹;

target\_ip: 下位机网络 IP;





target\_folder: 下位机目标文件夹位置;

target\_hostname: 下位机名称;

select\_file: 选择文件;

remote\_send\_file: 远程发送文件。

### 3.4.7 远程获取文件

| ▼ get_file      |                     |   |
|-----------------|---------------------|---|
| target_ip       | 192.168.1.102       |  |
| target_folder   | robot/map/data/gric |  |
| target_hostname | robot               |  |
| host_folder     | /home/l/map         |  |
| remote_get_file |                     |   |

target\_ip: 下位机网路 ip;

target\_folder: 下位机文件夹位置;

target\_hostname: 下位机名称;

host\_folder: 本机获取文件存放位置;

remote\_get\_file: 发送远程获取文件命令。

## 四、结束语

本文档是具有无线充电的变电站巡检机器人应用软件，是无线充电巡检机器人资料的第四册。文档不足之处恳请指正。