

车载软件开发基础

松灵小车任务4



**2023至2024学年第 1 学期**

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 团队作用 |
| 202324131109 | 阳沐云 | 主要负责完成任务（1）：编程实现文本识别,协作完成任务（2），撰写报告。 |
| 202324131117 | 李文鑫 | 主要负责完成任务（2）：编写程序，实现导航,协作完成任务（1），撰写报告。 |
| 任课教师 | 刘骥 | |
| 成 绩 |  | |

|  |  |
| --- | --- |
| 任务书 | |
| 任务内容 | 实验分组进行，每组人数不超过4人。在松灵小车上编写ROS程序，实现如下功能：   1. 采用gmapping方法进行SLAM建图； 2. 采用move\_base方法从固定起点到固定终点的小车自主导航； 3. 在环境实时变化的场景中，尽量保证更快更准确地进行小车自主导航； 4. 通过识别任务卡上的图标，理解车辆的目的地，并完成（3） |
| 程序规范 | （1）所有程序代码采用C++编写，使用git进行源代码管理；  （2）类名、变量名、函数名应符合C++的命名规范，并在代码中前后保持一致；  （3）涉及面向对象的程序，例如自定义的类，应符合面向对象的设计原则；  （4）正确使用头文件和源文件，自定义的头文件应符合头文件的编写原则，例如用条件宏定义确保头文件不被多次引用、不在头文件中进行类和函数的实现（模板除外）；  （5）项目必须是ROS项目，符合ROS的项目的规范，正确编写CmakeLists.txt等文件；  （6）程序能够在松灵小车上运行。 |
| 报告要求 | （1）报告至少应该包括人员分工、需求分析、程序设计、程序效果展示、总结分析4个部分；  （2）人员分工介绍组员各自的工作情况；  （3）需求分析侧重描述程序所需要实现的功能，功能预期的效果；  （4）程序设计描述组成程序的模块、类、函数以及他们之间的相互关系，若有算法，可以描述算法流程；  （5）程序效果展示除了程序运行效果截图之外，应该有必要的文字说明；  （6）总结分析可以分析实现的效果与理想情况的差异，分析导致这些差异的原因，切忌不要写成心得体会；  （7）报告应该格式规范、排版整洁、少语病和错误。 |
| 作业提交 | （1）含有git仓库（有.git目录）的完整源代码；  （2）程序功能演示的讲解视频；  （3）任务报告。 |
| 评分标准 | 按照五级制打分，分为优秀、良好、中等、及格、不及格，各评分项占总成绩的比例为：  （1）任务完成情况占评分的60%；  （2）程序规范占评分的20%；  （3）报告占评分的20%。  评分老师根据各部分的完成情况，直接给出总成绩。 |

**一、人员分工**

**阳沐云：**按照分工，主要负责完成任务

（1）tesseract.cpp程序的开发和实现。

（2）设计和编写相机图像的订阅逻辑，包括接收和转换相机图像数据。

（3）集成和使用Tesseract库进行文本识别。

（4）实现文本识别结果的修正和封装，并发布到ROS话题。

**李文鑫：**按照分工，主要负责完成任务

（1）负责goal\_publisher.cpp程序的开发和实现。

（2）设计和编写订阅文本识别结果的逻辑，包括接收和处理文本识别结果消息。

（3）创建导航目标点消息，并发布到ROS话题。

（4）集成和使用导航行为客户端，实现导航目标点的发布和导航任务的执行逻辑。

**二、需求分析**

本次任务一共编写了两个cpp文件，完成需要实现的功能。

1. **show\_speed.cpp：将相机捕获的图像进行文本识别并发布**
2. **捕获相机图像：**从ROS话题订阅相机图像数据。
3. **图像转换：**将ROS图像消息转换为OpenCV图像格式，以便进行文本识别。
4. **文本识别：**使用Tesseract库对图像中的文本进行识别。
5. **结果修正：**去除识别结果中的空格和不可见字符。
6. **结果发布：**将修正后的文本识别结果发布到ROS话题。
7. **goal\_publisher.cpp：发布导航目标点并执行导航：**
8. **订阅文本识别结果：**从ROS话题订阅文本识别结果，以获取导航目标点信息。
9. **发布导航目标点：**将确定的导航目标点发布到ROS话题，供导航系统使用。
10. **执行导航任务：**使用导航行为客户端执行导航任务，实现自主导航功能。

**三、程序设计（组成程序的模块、类、函数和关键算法）**

**① tesseract.cpp：将相机捕获的图像进行文本识别**

该程序主要用于处理相机图像并进行文本识别。以下是程序的简要介绍：

**所用函数：**

1. imageCallback函数：该函数是相机图像的回调函数，接收相机图像消息，并进行文本识别的处理。主要步骤包括：

* 将ROS图像消息转换为OpenCV图像格式。
* 使用Tesseract库在OpenCV图像上进行文本识别。
* 修正识别结果中的空格和不可见字符。
* 打印识别结果。
* 根据识别结果发布相应的文本消息。
* 释放Tesseract资源。

**关键算法：**

1. 图像处理：使用ROS提供的cv\_bridge库将ROS图像消息转换为OpenCV图像格式，方便后续的文本识别处理。
2. 文本识别：使用Tesseract库进行文本识别。首先初始化Tesseract实例并设置识别语言为英语。然后将OpenCV图像数据传递给Tesseract，并调用Recognize函数进行识别。最后通过调用GetUTF8Text函数获取识别结果。

tesseract.cpp发布识别结果：在tesseract.cpp中，识别结果通过全局变量text\_publisher发布到话题/text\_recognition/result上。在识别完成后，将识别结果封装成std\_msgs::String消息，并通过text\_publisher发布到话题上。

**② goal\_publisher.cpp: 发布导航目标点并执行导航**

该程序用于发布导航目标点并执行导航。以下是程序的简要介绍：

**所用函数：**

1. void textCallback(const std\_msgs::String::ConstPtr& msg)：识别结果的回调函数，用于处理识别结果。
2. void publishAndNavigateGoal(ros::Publisher& pub, MoveBaseClient& ac, double x, double y)：发布导航目标点并执行导航的函数。

**关键算法：**

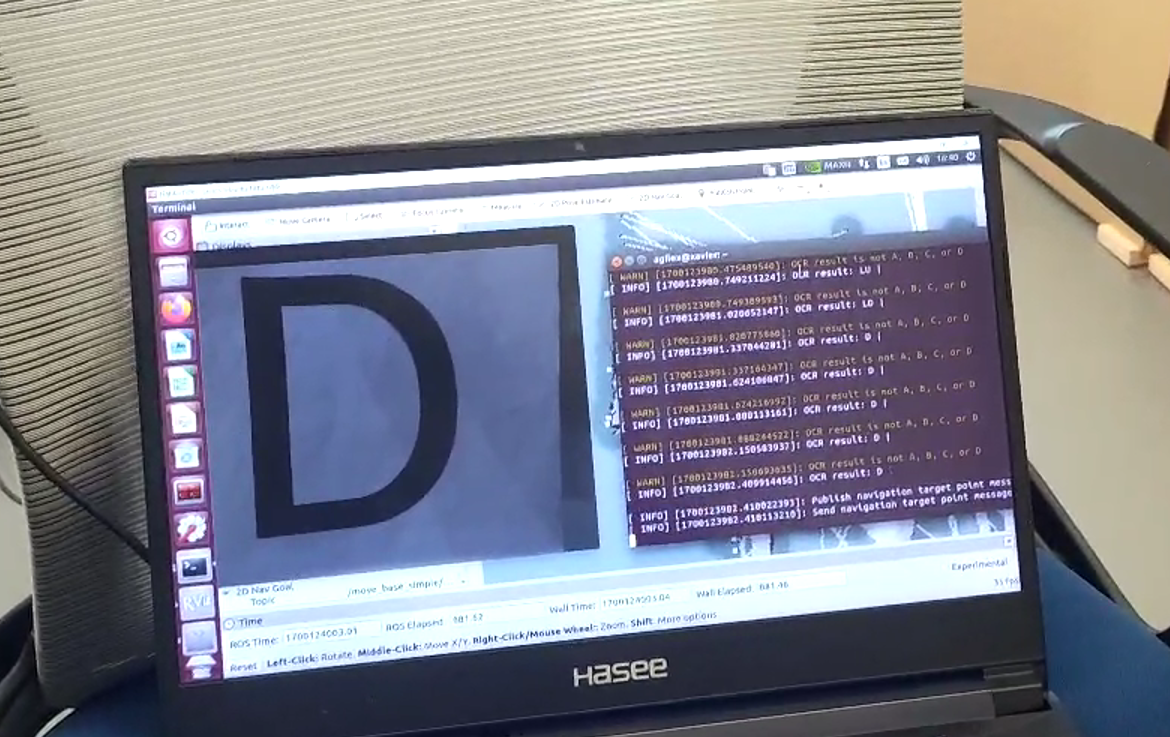
1. 导航目标点发布和执行：通过创建一个geometry\_msgs::PoseStamped消息，设置导航目标点的位置和朝向信息，并将该消息发布到话题(/move\_base\_simple/goal)上。然后使用actionlib库创建一个导航行为客户端，将导航目标点消息发送给move\_base服务器，并等待导航完成。

goal\_publisher程序订阅识别结果：goal\_publisher程序中的textCallback函数订阅了话题/text\_recognition/result，并在接收到消息时触发执行。通过订阅这个话题，goal\_publisher程序可以获取到tesseract.cpp发布的识别结果。

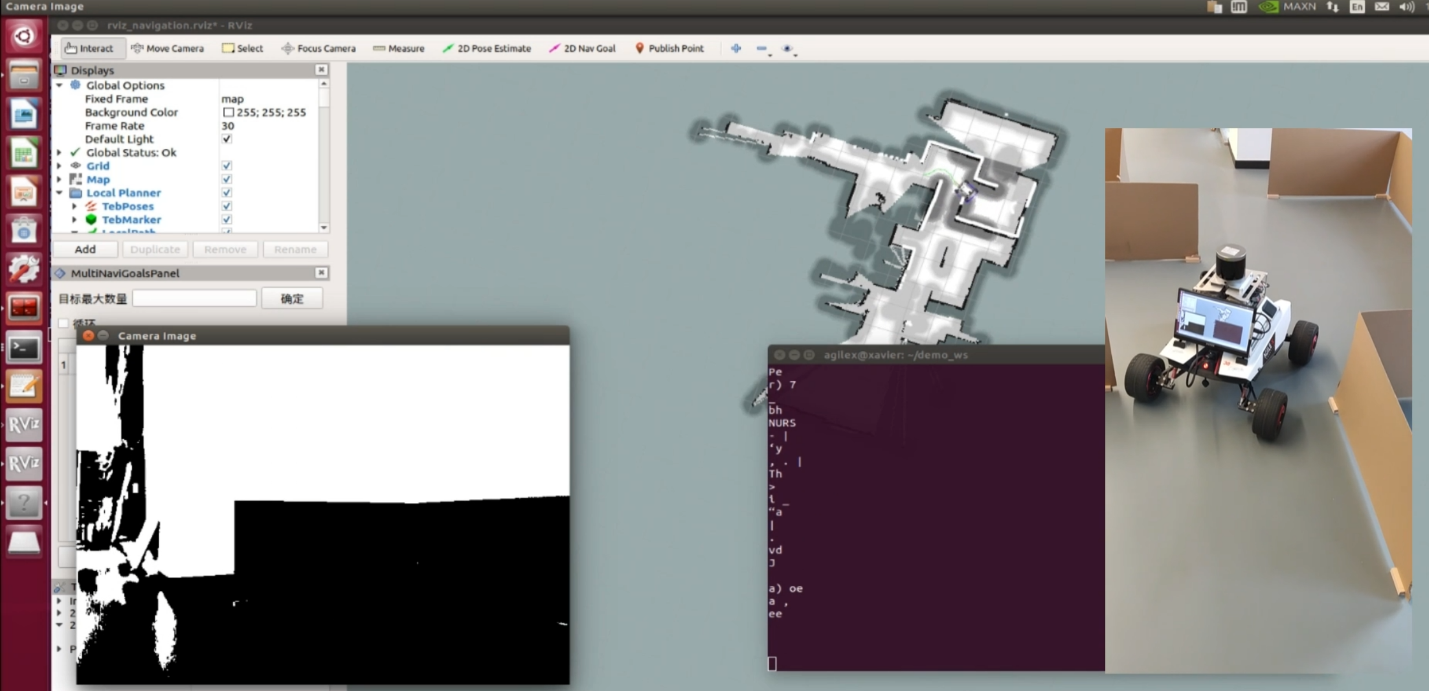
goal\_publisher程序根据识别结果执行导航：在goal\_publisher程序中，根据接收到的识别结果，调用publishAndNavigateGoal函数发布导航目标点并执行导航。识别结果作为参数传递给publishAndNavigateGoal函数，根据不同的识别结果确定导航目标点的位置坐标，然后通过发布者pub发布导航目标点消息，最终使用导航行为客户端ac执行导航。

**四、程序效果展示**

**1)** **将相机捕获的图像进行文本识别并发布**



**2)** **发布导航目标点并执行导航**



**五、总结和分析**

**分析**

在任务过程中，我们发现以下关键点：

1. 任务涉及两个C++程序，分别用于文本识别和导航目标发布与执行。
2. 使用ROS作为通信框架，通过发布者和订阅者实现消息传递。
3. 在文本识别程序中，使用OpenCV和Tesseract库对相机图像进行文本识别，并发布识别结果。
4. 在导航目标发布与执行程序中，接收文本识别结果，根据不同的识别结果发布相应的导航目标点，并执行导航任务。
5. 通过ROS节点的主循环，实现消息的处理和频率控制。
6. 程序中使用了ROS提供的库函数和消息类型，如ros::NodeHandle、ros::Publisher、ros::Subscriber等。
7. 程序中使用了ROS服务端和客户端的通信模式，如通过导航行为客户端发送导航目标点消息并等待导航完成的回应。
8. 通过ROS日志函数ROS\_INFO和ROS\_WARN输出程序运行状态和结果信息。

**结论**

本次实验成功完成了在ROS系统下的文本识别和导航任务。通过tesseract.cpp程序实现了相机图像上的文本识别，并将识别结果发布到/text\_recognition/result话题上。而goal\_publisher.cpp程序则实现了根据文本识别结果发布导航目标点，并执行导航任务的功能。通过订阅/text\_recognition/result话题获取识别结果，并根据不同的结果发布相应的导航目标点消息，利用导航行为客户端执行导航任务。这些功能的实现依赖于ROS的消息传递机制和相关库函数的使用。通过本次任务的实现，验证了在ROS系统下使用C++编程实现复杂任务的可行性