第一节课习题

题目一：熟悉Linux

1 .1apt-get软件安装的步骤，以及ubuntu管理软件依赖关系和软件依赖关系和软件版本的方式。

1.1.1 apt-get安装软件的整体步骤

使用 sudo apt-get install xxx 之后的操作可以分为四步：

第一步：扫描存放于本地的软件包列表，（该列表是由 apt-get update 刷新的，其意义是所绑定的软件源中所含有的软件）该步骤的目的是确定所指定的软件源中含有 xxx 软件，并找到最新的版本。软件源的设置可以在 软件和更新之中设置 看CSDN中有建议设置为国内的源方便下载的建议。

第二步：在找到所需的软件包之后，还需要检查软件包的依赖关系，确定支持该软件正常运行所需要的所有软件包。

第三步：从指定的软件源下载所需要的软件包

第四步：完成软件的安装，包括软件包的解压和自动完成应用程序的安装和配置

1.1.2 ubuntu管理软件依赖关系的方式

软件包依赖冲突的个人理解：复杂功能的建立，得益于简单的工具的组合应用。在软件的设计过程中也是如此，在编写软件不免会调用第三方的工具，而这些第三的工具库（函数库）也会有不同的版本，在多个软件都需要调用这个工具库时，不同的软件会对这个三方库的版本作出要求，此时就会容易发生依赖环境冲突。

ubuntu采用的是deb软件管理工具，这个软件会从软件源库里调用安装所需的安装包，而且可以自动分析和解决依赖关系，并将所有依赖的软件都安装为合适的版本。

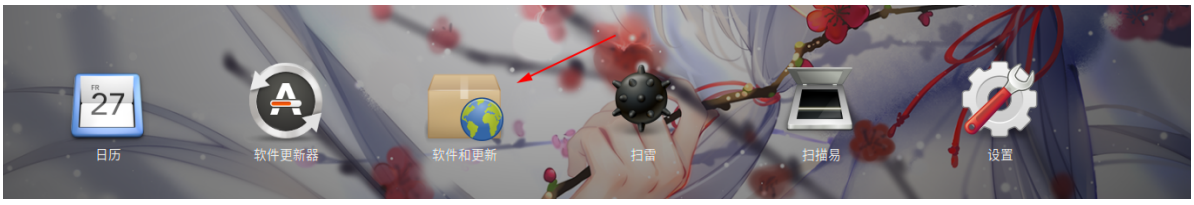
1.2什么是软件源？如何更换系统自带的软件源？如何安装来自第三方软件源中的软件？

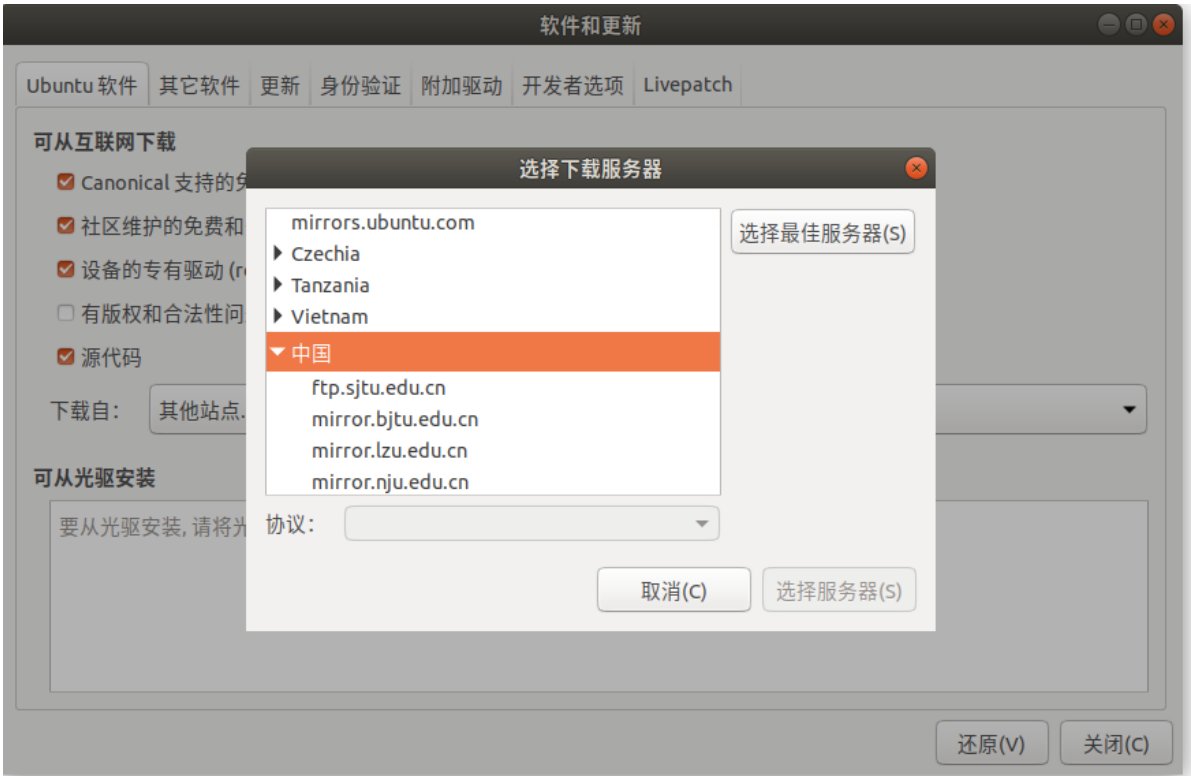
1.2.1什么是软件源

软件源是指debian系操作系统的应用程序安装包仓库，其中存放着大量的软件包，使用apt-get可以从软件源中下载软件。

1.2.2如何更换系统自带的软件源

方法一：使用图形界面





方法二：使用命令行

```
sudo gedit /etc/apt/sources.list
//打开软件源的设置文件，之后就可以添加自己想要加入的软件源 例如：deb
https://mirrors.ustc.edu.cn/ubuntu/ bionic main restricted universe multiverse 保存后退出执行下列命令
sudo apt-get update
```

1.2.3如何安装来自第三方软件源中的软件

安装第三方的软件时，只需下载对应的deb软件安装包，就可以根据引导完成安装。在完成安装后可能会有一些配置步骤，都可以根据deb提供者给出的参考进行操作。例如搜狗拼音就有全面详细的引导指南。

1.3除了apt-get以外，还有什么方式在系统中安装所需软件？除了ubuntu以外，其他发行版使用什么软件管理工具？请至少各列举两种。

1.3.1除了apt-get的安装方式还有

deb格式文件离线安装

cmake工程安装

pip install安装

1.3.2其他发行版使用的软件管理工具

Fedora的软件管理机制为RPM,使用的指令为rpm,rpmbuild

Archlinux使用pacman进行软件包管理

1.4环境变量PATH是什么？有什么用途？LD_LIBRARY_PATH是什么？指令ldconfig有什么用途?

1.4.1环境变量PATH是什么？有什么用途?

环境变量PATH：系统通过PATH这个环境变量来获得可执行文件所在的位置，然后运行对应的可执行文件

个人理解用途：对输入的命令到PATH中去检索，找到相关的可执行文件即告知命令对应的可执行文件所在的地址

1.4.2LD_LIBRARY_PATH是什么?

程序已经成功编译并且链接成功后，可使用LD_LIBRARY_PATH来搜索目录，该变量中只有动态库具有意义

1.4.3 ldconfidg有什么用途?

ldconfig是一个动态链接库管理命令，为了让动态链接库为系统所共享，还需要运行动态链接库的管理命令 -ldcongif。ldconfig 主要是在默认的搜索目录以及动态库配置文件/etc/ld.so.conf内所列的目录下，搜索出课共享的动态链接库（lib.so），进而穿检出动态装入程序（ld.so）所需要的链接和缓存文件。缓存文件默认为/etc/ld.so.cache，此文件保存已排好序列的的动态链接库名字列表。

1.5 Linux文件权限有哪几种？如何修改一个文件的权限?

Linux下的文件权限分为三种：r(读)，w(写)，x(执行)

使用chmod修改用户权限

```
chmod [ u / g / o / a ] [ + / - / = ] [ r / w / x ] file
```

该指令有两种使用方式

scheme1:

- [u / g / o / a] 为权限范围，其中
  u：User，即文件或目录的拥有者
  g：Group，即文件或目录的所属群组
  o：Other，除了文件或目录拥有者和所属群组外，其他用户都属于这个范围
  a：All，即全部用户

- 权限操作
  +表示增加权限
  -表示取消权限
  =表示取消之前的权限，并给予唯一的权限

- 权限代号
  r：读取权限，数字代号为 “4”
  w：写入权限，数字代号为 “2”
  x：执行权限，数字代号为 “1”
  -：不具备任何权限，数字代号为 “0”

例如：

```
sudo chmod u+rw /code/readme.txt
```

该指令的意思为，给文件拥有者增加读写操作

scheme2:

```
chmod [xyz] file
```

其中x，y，z分别指定User、Group、Other的权限；用三位二进制数表示 ” r , w , x”（注意顺序）三种权限，其中 0 代表没有该权限，1 代表有该权限，如 100 则表示，有 ‘r”权限，无 “w x”权限；再将这个三位的二进制数转为十进制，则是 x (或y，z)的值
例如

```
sudo chmod 777
```

表示所有用户都对该文件有读写执行权限

1.6 Linux 用户和用户组是什么概念？用户组的权限是什么意思？有哪些常见的用户组？

Linux操作系统是多用户的分时操作系统，具有功能强大的用户管理机制，它将用户分为组，每个用户都属于某个组，每个用户都需要进行身份验证，同时用户只能在所属组所拥有的权限内工作，这样不仅方便管理，而且增加了系统的安全性。

用户：分为普通用户、管理员用户（root用户）和系统用户。普通用户在系统上的任务是进行普通的工作，root用户对系统具有绝对的控制权，但操作不当会对系统造成损毁。所以在进行简单任务是进行使用普通用户。

用户组：用户组是用户的容器，通过组，我们可以更加方便的归类、管理用户。用户能从用户组继承权限，一般分为普通用户组，系统用户组，私有用户组。当创建一个新用户时，若没有指定他所属于的组，系统就建立一个与该用户同名的私有组。当然此时该私有组中只包含这个用户自己。标准组可以容纳多个用户,若使用标准组,在创建一个新的用户时就应该指定他所属于的组。

1.7 常见的Linux下C++编译器有哪几种？在你的机器上，默认用的是哪一种？它能够支持C++的那个标准？

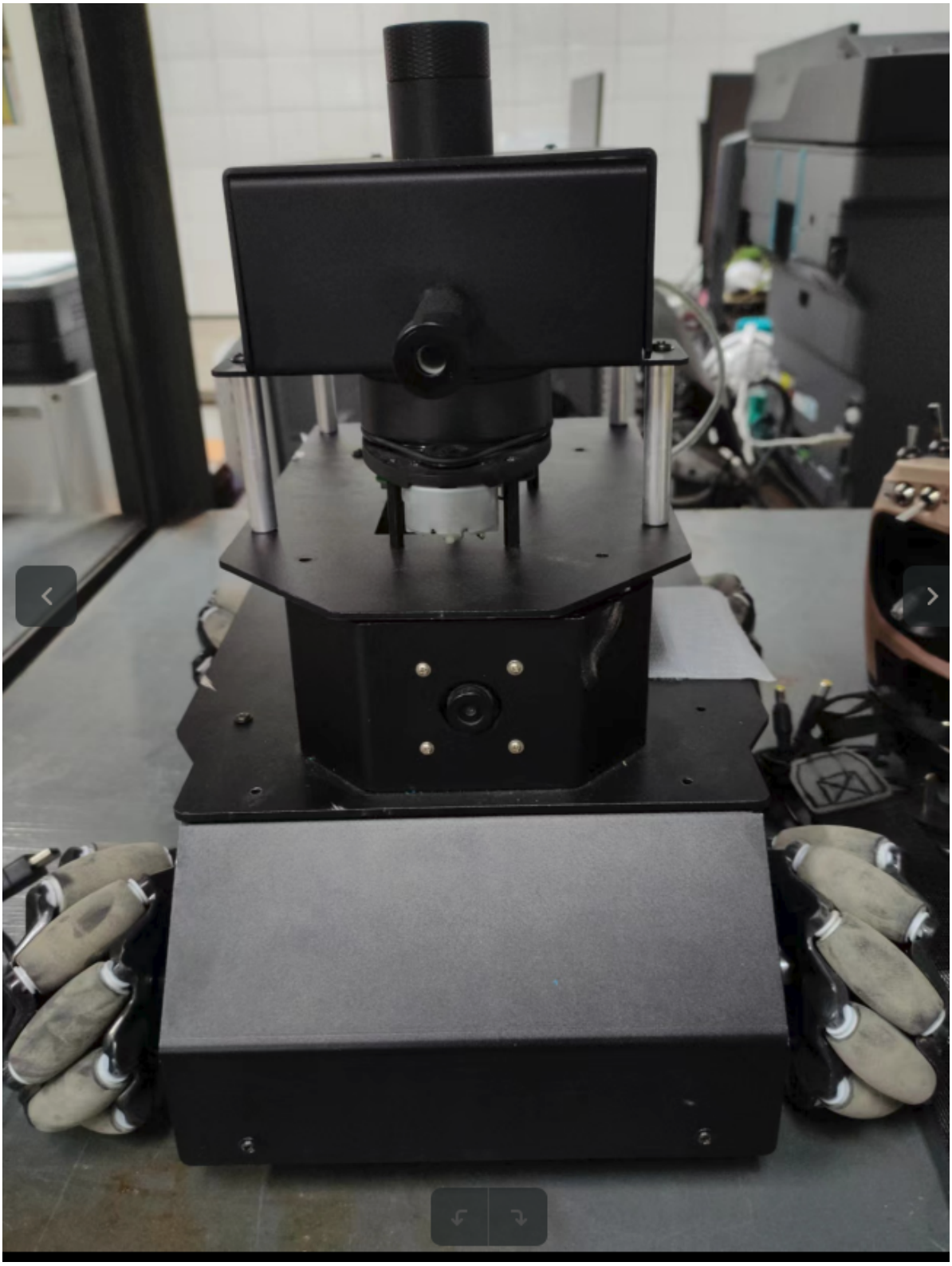
常见的c++编译器有 gcc和g++
我平时都用VScode进行简单的c++程序调试，支持大部分的c++标准

题目二：SLAM综述文献阅读

2.1 SLAM会在那些场合中用到？至少列举三个方向。

SLAM作为环境感知技术，可以用来给汽车自动驾驶做路径规划提供支撑。

SLAM可以用给机器人移动做技术支撑 如扫地机器人 以及采用激光雷达建图比赛用的小车



室内的物流分拣 AGV

除此之外SLAM还可以应用与AR/VR领域，今天也是期待刀剑神域降临的一天

2.2 SLAM 中定位与建图是什么关系?为什么在定位的同时需要建图?

定位：机器人获取自身在环境中的位置，即指测量自声和环境中物体的距离

建图：机器人将周围环境制成地图信息，即记录下机器人周边的环境信息

SLAM的关键是在实时状态下，机器人可以在定位的同事建立环境地图。这样可以使得机器人在未知环境下，进行探索，并实时反馈各种突发情况，降低了自动机器人的环境应用限制。

2.3 SLAM 发展历史如何？我们可以将它划分成哪几个阶段?

2.3.1 发展历史

概率SLAM问题最早起源于1986年在旧金山举办的IEEE机器人自动化会议中。经过几年的深耕，Smith等人发表了具有里程碑意义的论文。在最早提出SLAM的一系列论文中，当时的人们称它为“空间状态不确定性的估计”。而SLAM这一概念最早由Hugh Durrant-Whyte 和 John J.Leonard提出。SLAM主要用于解决移动机器人在未知环境中运行时定位导航与地图构建的问题。

2.3.2 阶段划分

第一阶段：定位和建图分开研究的阶段

第二阶段：定位和建图在一起进行研究

第三阶段：开始运用摄像头的阶段（在此之前由于计算机视觉的发展限制，一直用的都是激光测距的方法）

第四阶段：多传感器融合和深度学习和SLAM的融合

2.4 从什么时候开始SLAM区分为前段和后端？为什么我们要把SLAM区分为前段和后端?

前端是进行特征提取，获取相邻帧的位姿变换，而后端则是进行地图的拼接和位姿优化。

在特定的机器人应用中，或许很难直接写出传感器的测量结果，如状态的分析函数，正如MAP估计中需要使用的。举个例子，如果未处理的传感器数据是一个图像，可能很难将每个像素的强度表达为SLAM的状态函数；对于更简单的传感器也会出现同样的困难（例如，一个单光束的激光）。在这两个情况中，该问题与我们无法设计出一个充分一般的表达式来表述环境的事实有关。甚至是一个十分一般的表达式，都很难写出该表达式的与测量有关的参数。

出于这个原因，在SLAM后端之前，通常需要有一个模块，称之为前端，可以从传感器数据中提取出相关的特征。例如，在视觉SLAM中，前端从环境中提取出几个可区分点的像素位置；观测到这些点的像素现在就可以很容易地在后端中建模。前端还负责将每个测量结果与环境中的特定地标（如三维点）相关联：因此也称之为数据关联。

分前后端的原因主要是前后端由于所需信息量、执行频率的不同导致无法将二者融合成一个模块。

所需信息量不同：前端主要通过相邻帧之间的特征对应关系获取相邻帧的位姿对应关系，而后端则是需要综合多帧的观测数据，对多个姿态和地图进行联合优化，消除前端带来的累计误差。

执行频率不同：前端由于考虑数据量较少，因此计算速度快，执行频率高，后端由于需要处理的数据多，过程非线性，因此计算速率底，执行频率底。

2.5 列举三篇在SLAM领域的经典文献。

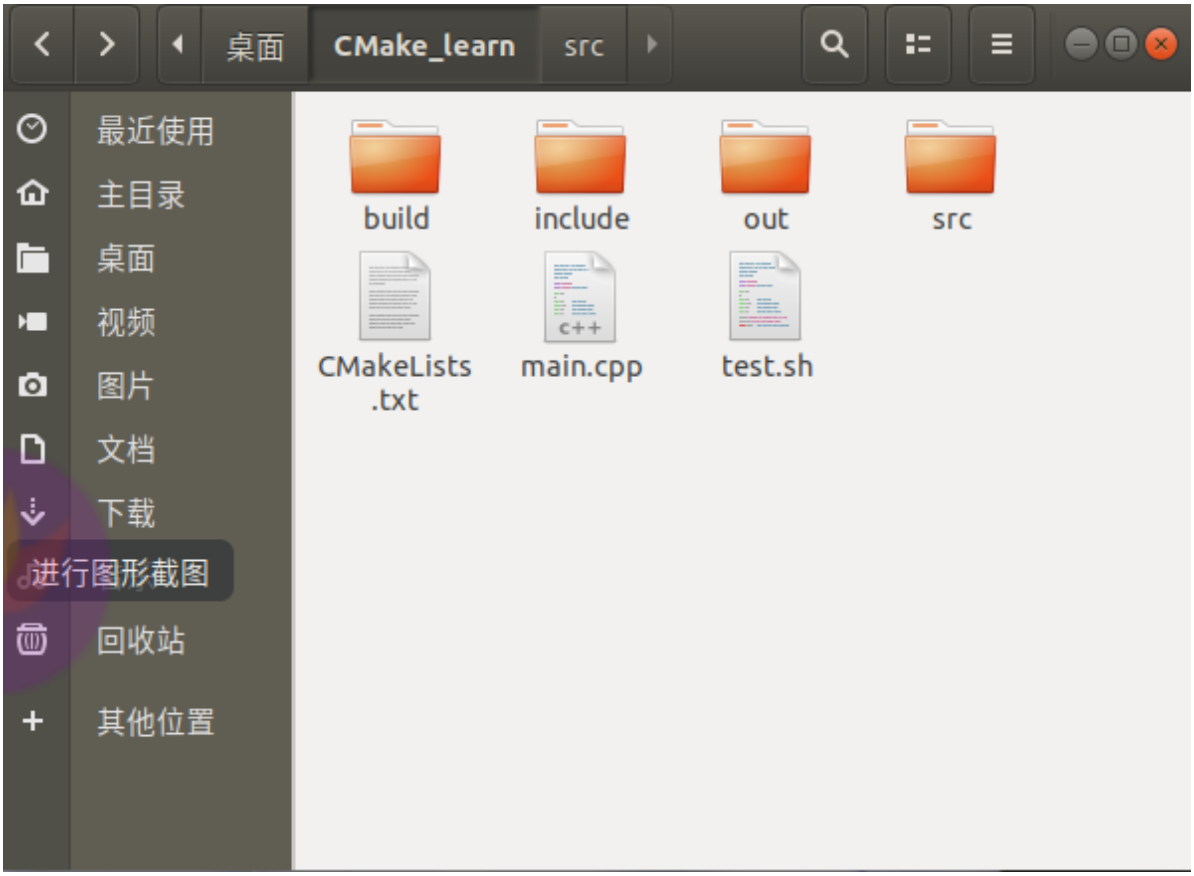
《LSD-SLAM: Large-Scale Direct Monocular SLAM》

《ORB-SLAM:ORB-SLAM: a Versatile and Accurate Monocular SLAM System》

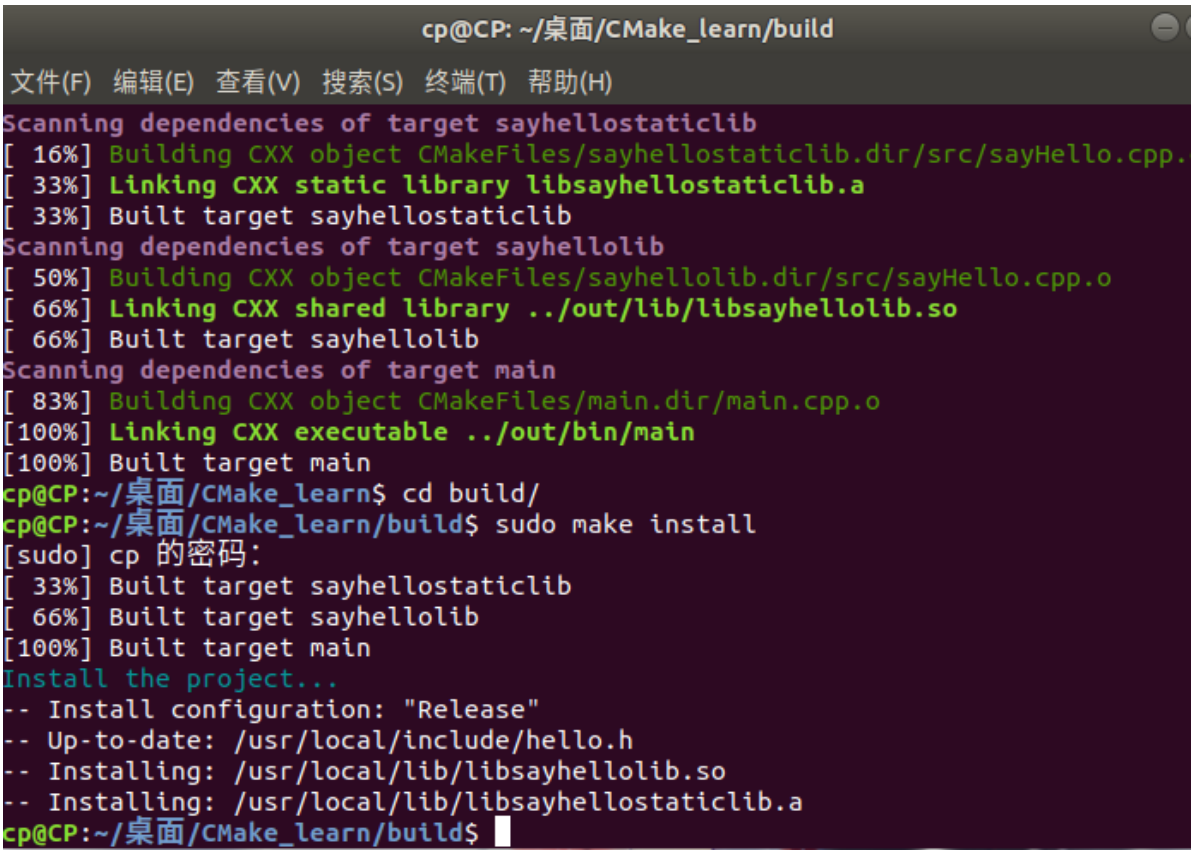
《VINS-Mono: A Robust and Versatile Monocular Visual Inertial State Estimator》

题目三：CMake练习

暂时还没有理解FindHello.cmake,现在完成的工作有



成功编译的截图



./include/hello.h

```
#include<iostream>

void  sayHello();
```

./src/hello.cpp

```
#include<iostream>

void sayHello()
{
   printf("Hello SLAM\n");
   std::cout<<"Hello SLAM"<<std::endl;
}
```

main.cpp

```
#include<iostream>
#include"include/hello.h"

int main(int argc,char ** argv){


sayHello();
return 0;

}
```

CMakeLists.txt

```
cmake_minimum_required(VERSION 2.8)

project(hello)

set(CMAKE_BUILD_TYPE Release)

set(CMAKE_RUNTIME_OUTPUT_DIRECTORY ${PROJECT_SOURCE_DIR}/out/bin)
set(CMAKE_LIBRARY_OUTPUT_DIRECTORY ${PROJECT_SOURCE_DIR}/out/lib)

set(CMAKE_MODULE_PATH ${PROJECT_SOURCE_DIR})



add_library(sayhellolib SHARED src/sayHello.cpp)
```

```
add_library(sayhellostaticlib  src/sayHello.cpp)

add_executable(main main.cpp)

target_link_libraries(main sayhellolib)


#install
set(CMAKE_INSTALL_PREFIX /usr/local)

install(FILES ${PROJECT_SOURCE_DIR}/include/hello.h DESTINATION include)

install(TARGETS sayhellolib sayhellostaticlib
LIBRARY  DESTINATION lib
ARCHIVE  DESTINATION lib
)
```

test.sh

```
mkdir build
cd build
cmake ..
make
sudo make install
cd ../out/bin/main
```

题目四：

4.1通过命令行安装

```
sudo apt-get install libgflags-dev
sudo apt-get install libgoogle-glog-dev
sudo apt-get install libgtest-dev
```

虽然用的命令行安装，但是我也不知道什么时候电脑就已经装了新版本的了。。。

这个，这的还没有接触过，看看后补

题目五：

5.1 完成ORB-SLAM2的源码下载

太慢了，我电脑上有好多个版本了，记忆中直接down下来的 会有很多报错。。。

目前用的是一个把常见错误调通了的，可以跑通的

5.2 通过阅读CMakeLists.txt理解orb-slam2

a) 在exemple中会生成一个RGBD两个stereo三个momocular一共六个可执行文件，会在lib中生成一个libORB_SLAM2.so的动态库文件

b)

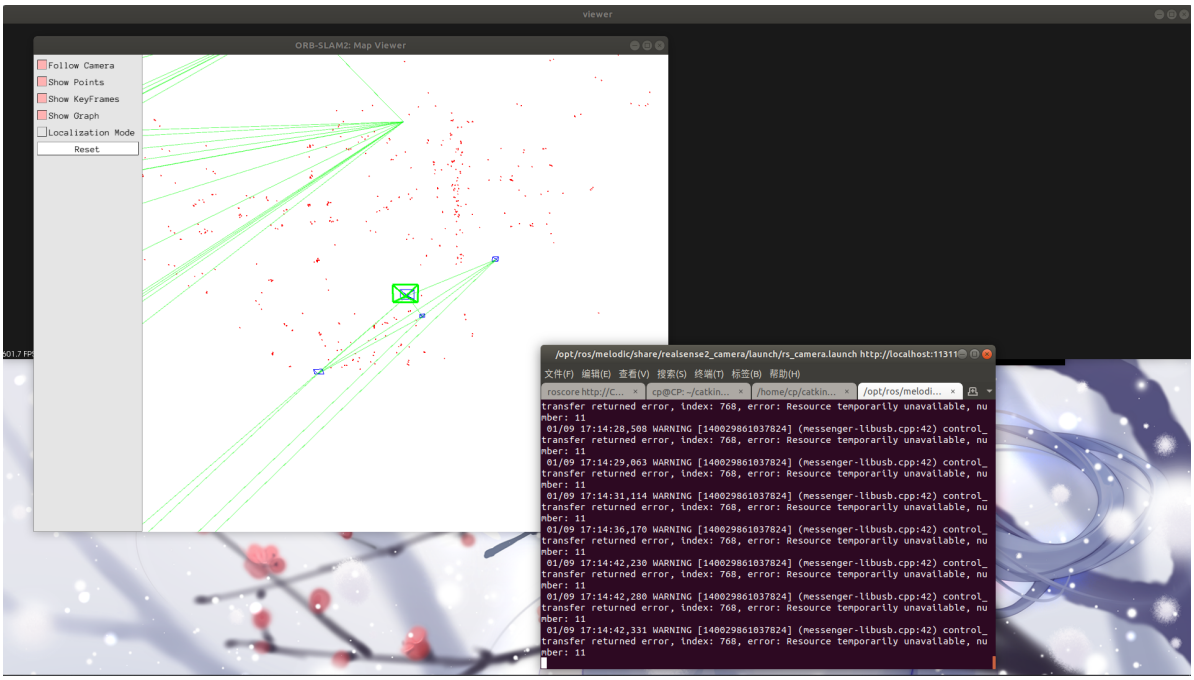
src: 动态库的（也是整个工程的）源码文件，一共十九个

include：各个源码文件对应的头文件

examples: momocular rgbd stereo 以及 在ROS环境下运行的示例文件

c)用到了六个库，分别是 工程的动态库 orb_slam2 opencv eigen pangolin Dbow2 g2o

题目六：



唉！这个代码拿到手很久了

一直想要用RGBD把稠密建图，还有八叉树图调试出来

但是一直不对，现在遇到的问题猜测是相机读取的参数不对

我准备用标定板，标定一下试试

感觉这个图截图就能看出相机不太对。。。