**第1章作业**

**用户名：MagicTZ**

**时间：19.09.2020**

**二 熟悉Linux**

1. **如何在Ubuntu中安装软件（命令行界面）？它们通常被安装在什么地方？**

* 利用命令行代码：sudo apt/apt-get install [software\_name] 安装软件，软件被安装在 /usr 的目录下面

1. **Linux的环境变量是什么？如何定义新的环境变量？**

* 按照Life cycle划分，可以分为永久环境变量和临时环境变量以及按照作用域划分，可以分为系统环境变量和用户环境变量。环境变量有：

PATH：指定命令的搜索路径

HOME：指用户的主工作目录

LOGNAME：指当前用户的登录名

HOSTNAME：指主机的名称

SHELL：指当前用户用的是哪种Shell

LANG/LANGUGE：语言相关的环境变量

MAIL：指当前用户的邮件存放目录

* 如何定义：
  + **所有用户（永久）**：在/etc/profile文件中添加变量（对所有用户生效），立马生效需要运行source/etc/profile，否则需要下次重进用户才能生效
  + **单一用户（永久）**：在用户目录下的.bash\_profile文件中添加变量，立马生效需要运行source ~/.bash\_profile，否则需要下次重进才能生效
  + **对当前shell（BASH）有效（临时）**：在shell的命令行下直接使用export 变量名=变量值

1. **Linux根目录下面的目录结构是什么样的？至少说出3个目录的用途。**

* /bin：系统放置执行文件的目录，放置了在单人维护模式下还能够被操作的指令，主要有：cat，chmod，chown，date，mv，mkdir，cp等
* /boot：这个目录主要在放置开机会使用到的档案，包括Linux核心档案以及开机选单与开机所需配置文件等
* /dev：在linux上，任何装置与借口设备都是以档案的形态存在于这个目录当中的。
* /etc：系统主要的配置文件几乎都是放置这个目录内，例如人员的账号密码文件
* /lib：系统的library文件
* /home：home directory
* /opt：给第三方协力软件放置的目录
* /sbin：重要的系统执行文件

1. **假设我要给 a.sh 加上可执行权限，该输入什么命令？**

* 利用chmod命令进行权限更改，如 【chmod a+x a.sh】

1. **假设我要给 a.sh 文件的所有者改成xiang：xiang， 该输入什么命令？**

* chown xiang:xiang a.sh

**三 SLAM 综述文献阅读**

1. **SLAM会在哪些场合中用到？至少列举三个方向。**

* Indoor navigation
* VR/AR device
* Indoor robot
* Autonomous driving

1. **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？**

* 互相依赖的关系，鸡生蛋，蛋生鸡。建图需要依赖于精确的位置描述，定位的精度也取决于建图的效果。
* 定位是核心，地图构建则不断地为定位提供参照标准，根据建图的结果，可以不断地对位置进行修正。同时，地图也能够用来估计机器人的姿态以及提供路径规划的信息。如果只进行单一的定位，是无法求解的。

1. **SLAM发展历史如何？我们可以将它划分成哪几个阶段？**

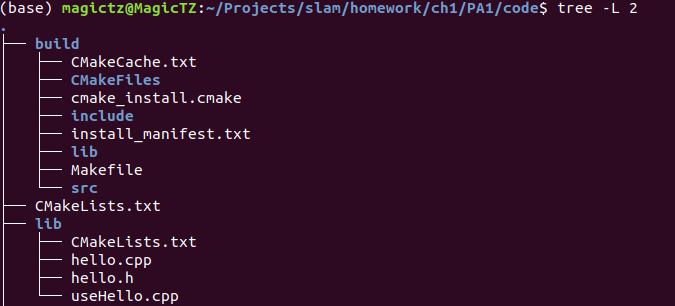
* 最开始，Mapping和localization是独立进行研究，并且最早是起源于机器人领域，在CV领域中相似的技术是SFM（Structure From Motion）。
* 后面，发现它们之间的相互联系，因此出现了SLAM（Simultaneous Localization and Mapping）
* 2005年，利用相机作为传感器，被称为visual SLAM
* 2011年，VIN SLAM（visual-inertial SLAM）出现
* 2015年，ORB-SLAM开源并成为当时性能最好的单目V-SLAM系统
* 近些年，由于激光雷达和深度学习方法的兴起，激光SLAM和基于深度学习的SLAM也成了非常重要的研究方向
* 阶段划分：
  + 1986-2004，classical age
  + 2004-2015，algorithmic-analysis age
  + 2015-至今

1. **列举三篇在SLAM领域的经典文献。**

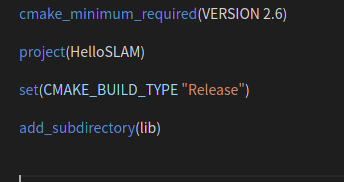
* Durrant-Whyte, H, and Bailey, Tim. ”Simultaneous Localization and Mapping: Part I.” IEEE Robotics & Amp Amp Automation Magazine 13.2(2006):99 - 110.
* Fuentes-Pacheco, Jorge, J. Ruiz-Ascencio, and J. M. Rendón-Mancha. ”Visual simultaneous localization and mapping: a survey.” Artificial Intelligence Review 43.1(2015):55-81.
* Davison, Andrew J., et al. ”MonoSLAM: Real-time single camera SLAM.” IEEE transactions on pattern analysis and machine intelligence 29.6 (2007): 1052-1067.

**四 CMake练习**

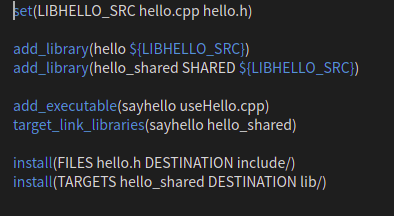
1. **文件的组织格式（以tree的形式）**



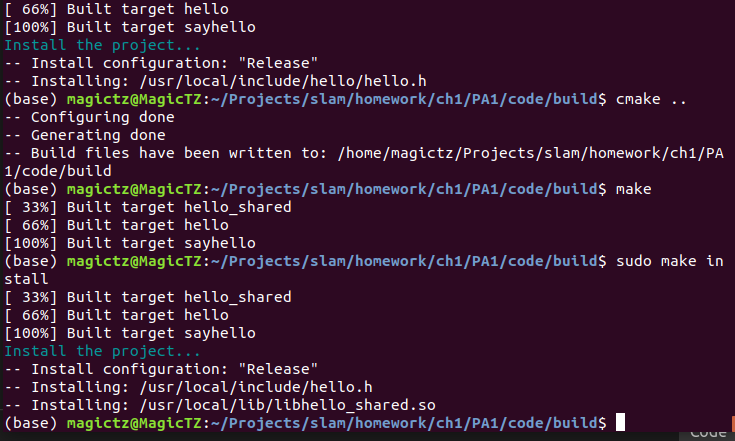
其中，主目录的CMakeLists.txt的内容如下：



Lib目录下的CMakeLists.txt的内容为：

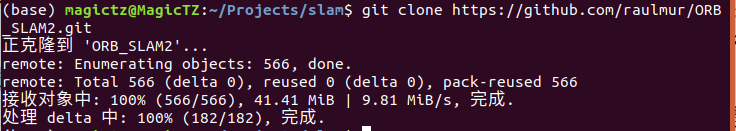


最后使用cmake编译安装得到的结果为：



**五 理解ORB-SLAM2框架**

1. **从github上下载ORB-SLAM2的截图**



1. **阅读ORB-SLAM2代码，回答问题**
   1. **ORB-SLAM2将编译出什么结果？有几个库文件和可执行文件？**

ORB-SLAM2使用已经写好的build.sh文件进行编译，首先对第三方库进行编译，然后对Vocabulary进行解压，最后对主体部分进行编译。

库文件：19+5（第三方库） 可执行：6 (rgbd\_tum, stereo\_kitti, stereo\_euro, mono\_tum, mono\_kitti, mono\_euroc)

* 1. **ORB-SLAM2 中的 include, src, Examples 三个⽂件夹中都含有什么内容？**

include: header feiles

src: .cc文件（Unix系统常用c++源文件）

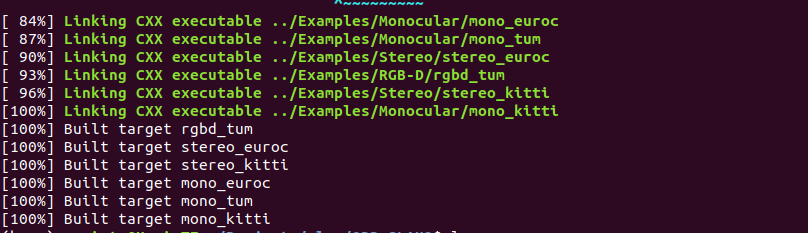
examples: 4个不同针对不同摄像头/场景的例子（Monocular, RGB-D, ROS, Stereo)

* 1. **ORB-SLAM2 中的可执⾏⽂件链接到了哪些库？它们的名字是什么？**

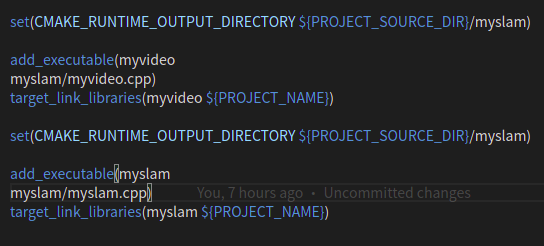
链接到了OpenCV\_LIBS, EIGEN3\_LIBS, Pangolin\_LIBRARIES, libDBoW2.so, libg2o.so

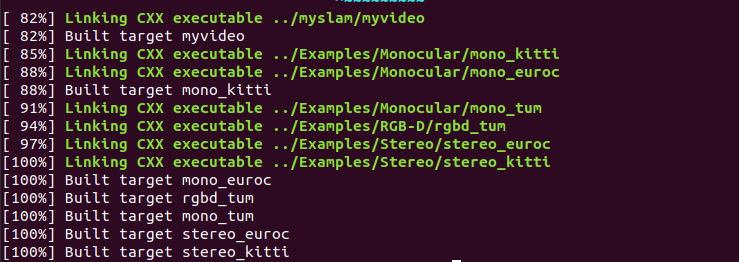
**六 使用摄像头或视频运行ORB-SLAM2**

1. **ORB-SLAM2编译截图**

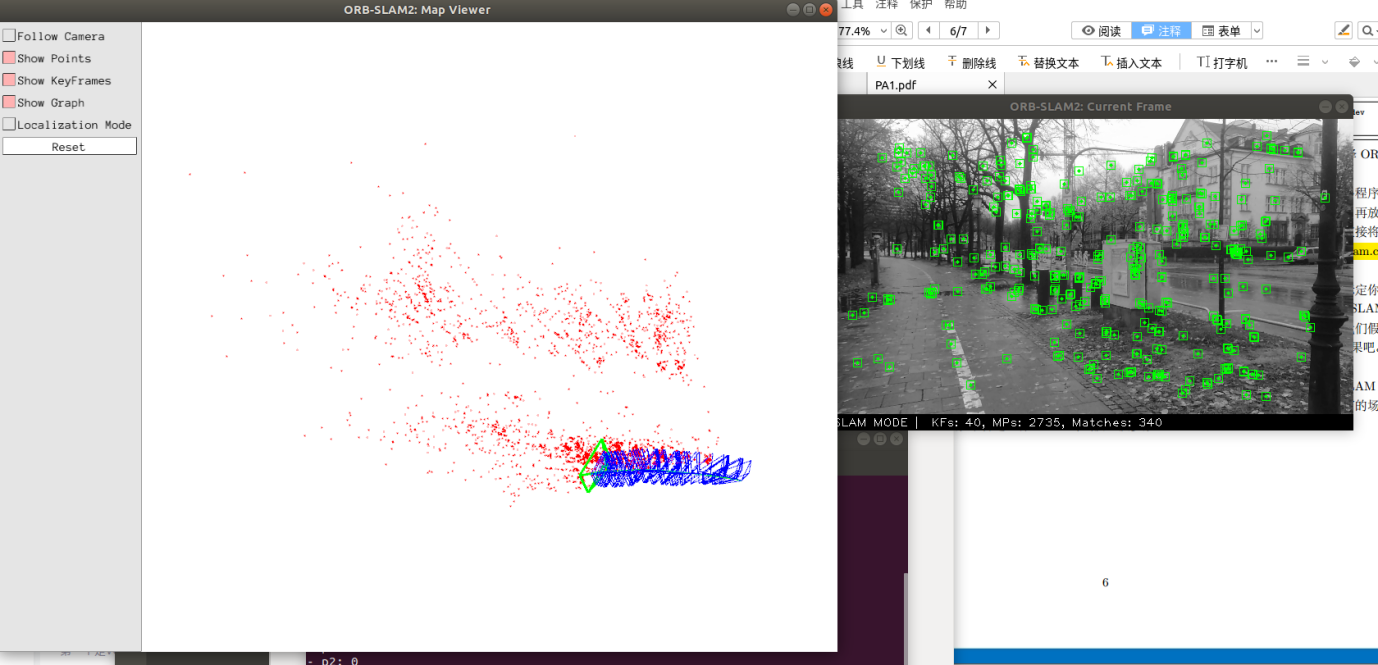


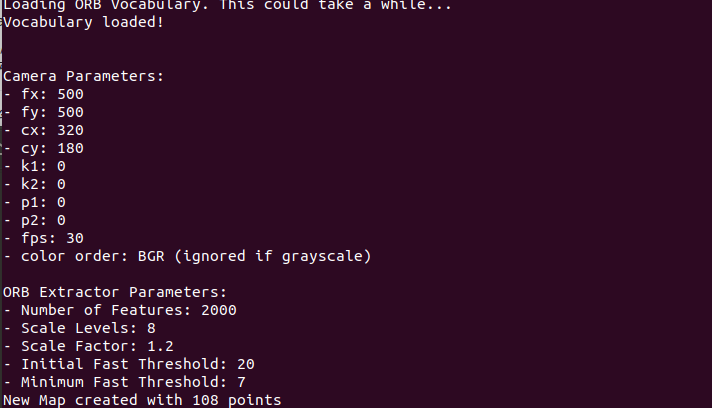
1. **将myslam.cpp或myvideo.cpp加入到ORB-SLAM2工程中，CMakeLists.txt修改方案为**：在CMakeLists.txt的尾部添加新的输出路径和可执行文件



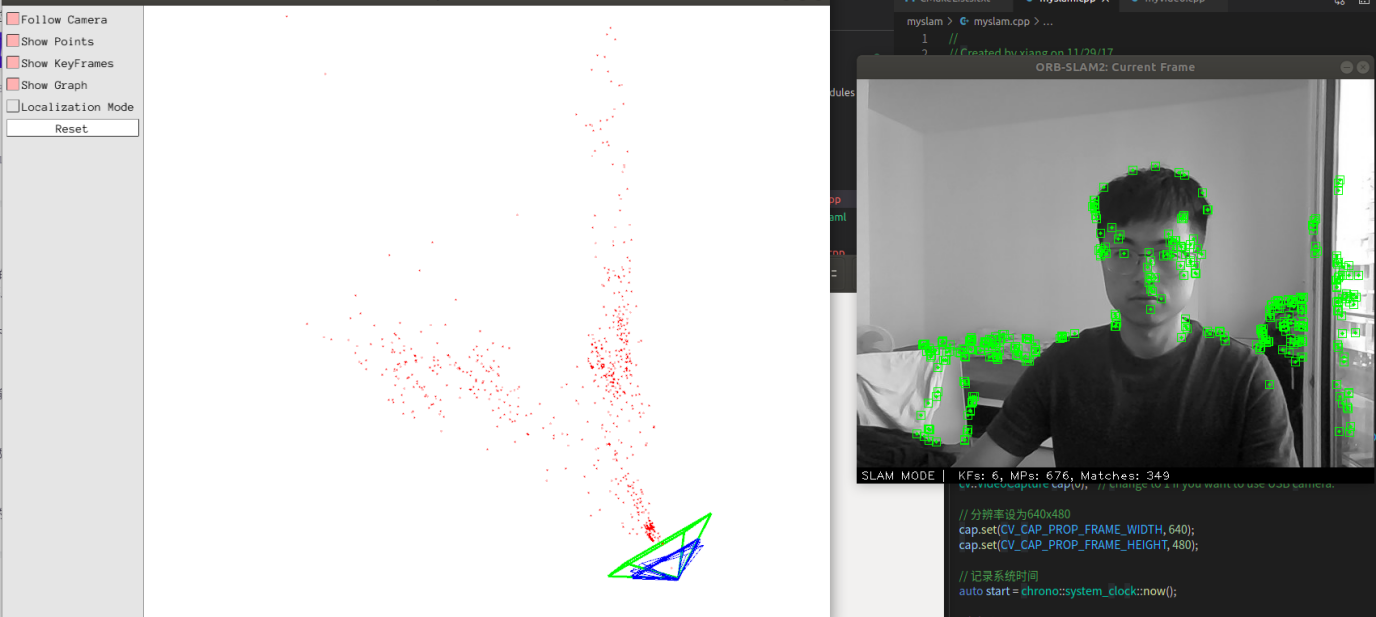
其cmake编译的结果如下：

1. **修改后的效果图**

第一种方案是利用已经拍摄好的视频作为输入进行Tracking，该种方案的可执行文件是myvideo，利用ORB-SLAM2对视频进行处理，示意图如下：



第二种方案是利用电脑的摄像头作为输入，进行特征点提取和Tracking，该种方案的可执行文件是myslam，示意图如下：



以上两种方案都没有对相机进行标定，因此只能说了解了Tracking的这个过程，而且由于都没有Loop，所以也没有观察到Loop Closure的过程。整个过程是先将前面几帧作为基准帧，然后给予前面几帧来首先确定一个相机的位姿，然后相机不断运动，并通过特征点提取和匹配来不断更新相机的位姿。