

基于ORB-SLAM3库搭建SLAM系统

作者：郭鸿源

转载请注明地址

原文地址：[📄 基于ORB-SLAM3库搭建SLAM系统](#)

一、基础部分：基于ORB-SLAM3库搭建SLAM系统

1. 注意事项



如果是新系统，需要先安装个git、vi编辑器、DBoW2、g2o（安装过程及可能遇到的问题、相应的对策解决方案放在附录中）

- 备注：DBoW2、g2o在安装ORB-SLAM3中会自动安装编译，不再在附录中另行安装
- DBoW2：主要用于回环检测
- g2o：用于图优化

2. 准备工作，下载安装要用的文件

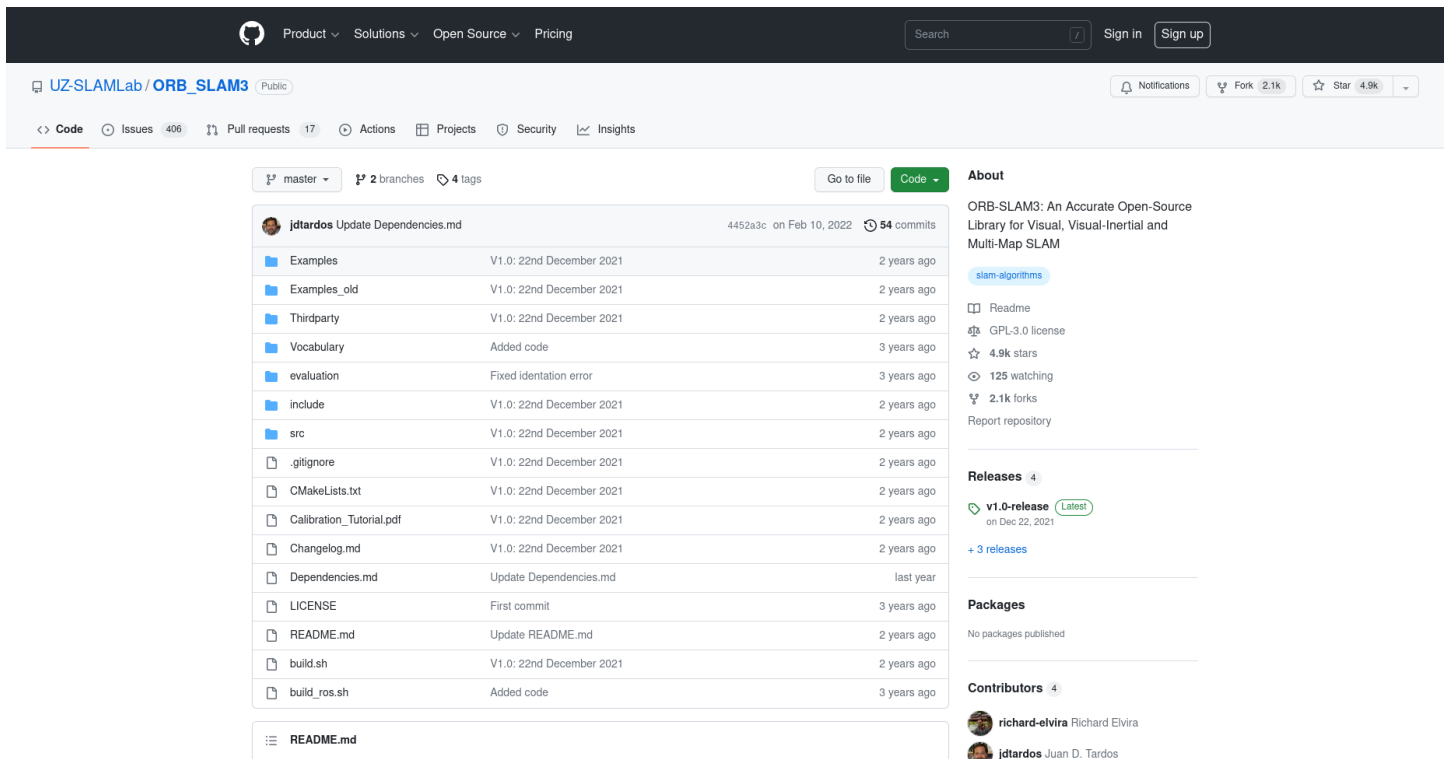
2.1 新建一个文件夹，用于存放下载的文件

- 1 这里我们将这个文件夹取名为 homework_exercise

2.2 在github上下载ORB-SLAM3库

在 homework_exercise 文件夹下打开终端，输入下列命令行或直接在github上下载提取

- 1 `git clone https://github.com/UZ-SLAMLab/ORB_SLAM3`



UZ-SLAMLab / ORB_SLAM3 Public

<> Code Issues 406 Pull requests 17 Actions Projects Security Insights

master 2 branches 4 tags

Go to file Code

About

ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM

slam-algorithms

Readme

GPL-3.0 license

4.9k stars

125 watching

2.1k forks

Report repository

Releases 4

v1.0-release Latest on Dec 22, 2021

+ 3 releases

Packages

No packages published

Contributors 4

richard-elvira Richard Elvira

jdtardos Juan D. Tardos

File/Folder	Commit Date	Age
Examples	V1.0: 22nd December 2021	2 years ago
Examples_old	V1.0: 22nd December 2021	2 years ago
Thirdparty	V1.0: 22nd December 2021	2 years ago
Vocabulary	Added code	3 years ago
evaluation	Fixed indentation error	3 years ago
include	V1.0: 22nd December 2021	2 years ago
src	V1.0: 22nd December 2021	2 years ago
.gitignore	V1.0: 22nd December 2021	2 years ago
CMakeLists.txt	V1.0: 22nd December 2021	2 years ago
Calibration_Tutorial.pdf	V1.0: 22nd December 2021	2 years ago
Changelog.md	V1.0: 22nd December 2021	2 years ago
Dependencies.md	Update Dependencies.md	last year
LICENSE	First commit	3 years ago
README.md	Update README.md	2 years ago
build.sh	V1.0: 22nd December 2021	2 years ago
build_ros.sh	Added code	3 years ago

README.md

2.3 下载Pangolin

在 homework_exercise 文件夹下打开终端，输入下列命令行或直接在github上下载提取

```
1 git clone https://github.com/stevenlovegrove/Pangolin.git
```

2.4 下载Eigen3

在 homework_exercise 文件夹下打开终端，输入下列命令行或直接在github上下载提取

```
1 git clone https://github.com/eigenteam/eigen-git-mirror
```

3. 安装下载的及相关所需的库

3.1 安装 Eigen3库

在 homework_exercise 文件夹下打开终端，输入下列命令行

```
1 cd eigen-git-mirror
2 mkdir build
3 cd build
4 cmake ..
5 sudo make install
6
```

3.2 安装 Pangolin

在 homework_exercise 文件夹下打开终端，输入下列命令行

3.2.1 安装Pangolin需要的依赖工具

在终端依次输入以下指令

```
1 sudo apt install libgl1-mesa-dev
2 sudo apt install libglew-dev
3 sudo apt install cmake
4 sudo apt install libpython2.7-dev
5 sudo apt install pkg-config
6 sudo apt install libegl1-mesa-dev libwayland-dev libxkbcommon-dev wayland-protoc
```

3.2.2 安装 Pangolin


在 homework_exercise 文件夹下打开终端，输入下列命令行

```
1 cd Pangolin
2 mkdir build
3 cd build
4 cmake ..
5 cmake --build .
```

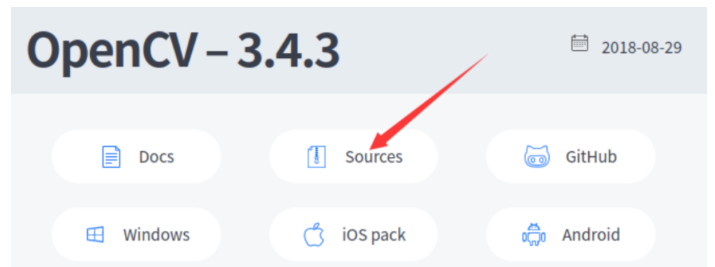
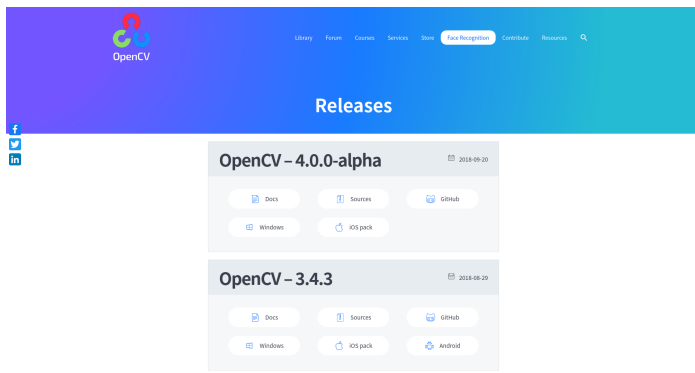
3.3 安装 OpenCV3.4.3

3.3.1 在官网下载OpenCV库

官网网址：<https://opencv.org/releases/page/5/>

 由于OpenCV - 4.0.0-alpha使用的博主不多，更常用的是3.4.3版本，这里笔者使用的是3.4.3

在“Opencv -3.4.3” 板块 中找到“source” 并点击



点击下载之后，将压缩文件放到 homework_exercise 文件夹，右键提取文件进行解压更新一下，准备下一步，安装OpenCV依赖库。在终端输入以下指令

```
1 sudo apt-get update
```

3.3.2 安装OpenCV依赖

在终端中输入下列指令

```
1 sudo apt-get install build-essential libgtk2.0-dev libavcodec-dev libavformat-dev libjpeg-dev libtiff5-dev libswscale-dev libjasper-dev
2
3 # 原博客安装的是libtiff4-dev，有博主运行过程报错，改成libtiff5-dev
4 # 这里，笔者使用安装libtiff4-dev时也没有报错，笔者最后选择libtiff5-dev安装。大家可以根据实际情况选择。
```

3.3.3 编译OpenCV

在 homework_exercise 文件夹下打开终端，输入下列命令行

```
1 cd opencv-3.4.3
2 mkdir build
3 cd build
4 cmake -D CMAKE_BUILD_TYPE=Release -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

再次执行cmake指令，完成编译任务，在终端中输入下列命令行

```
1 cmake -D CMAKE_BUILD_TYPE=Release -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

用make完成后续编译任务，在终端中输入下列命令行

```
1 make -j4
```

最后，完成编译安装

```
1 sudo make install
```

3.3.4 配置环境

3.3.4.1 添加库路径

```
1 sudo /bin/bash -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'
```

3.3.4.2 更新系统库

```
1 sudo ldconfig
```

3.3.4.3 配置bash

```
1 sudo gedit /etc/bash.bashrc
```

打开后在文件末尾添加下面两行代码

```
1 PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig  
2 export PKG_CONFIG_PATH
```

保存文件，执行下面的指令使配置生效

```
1 source /etc/bash.bashrc
```

更新

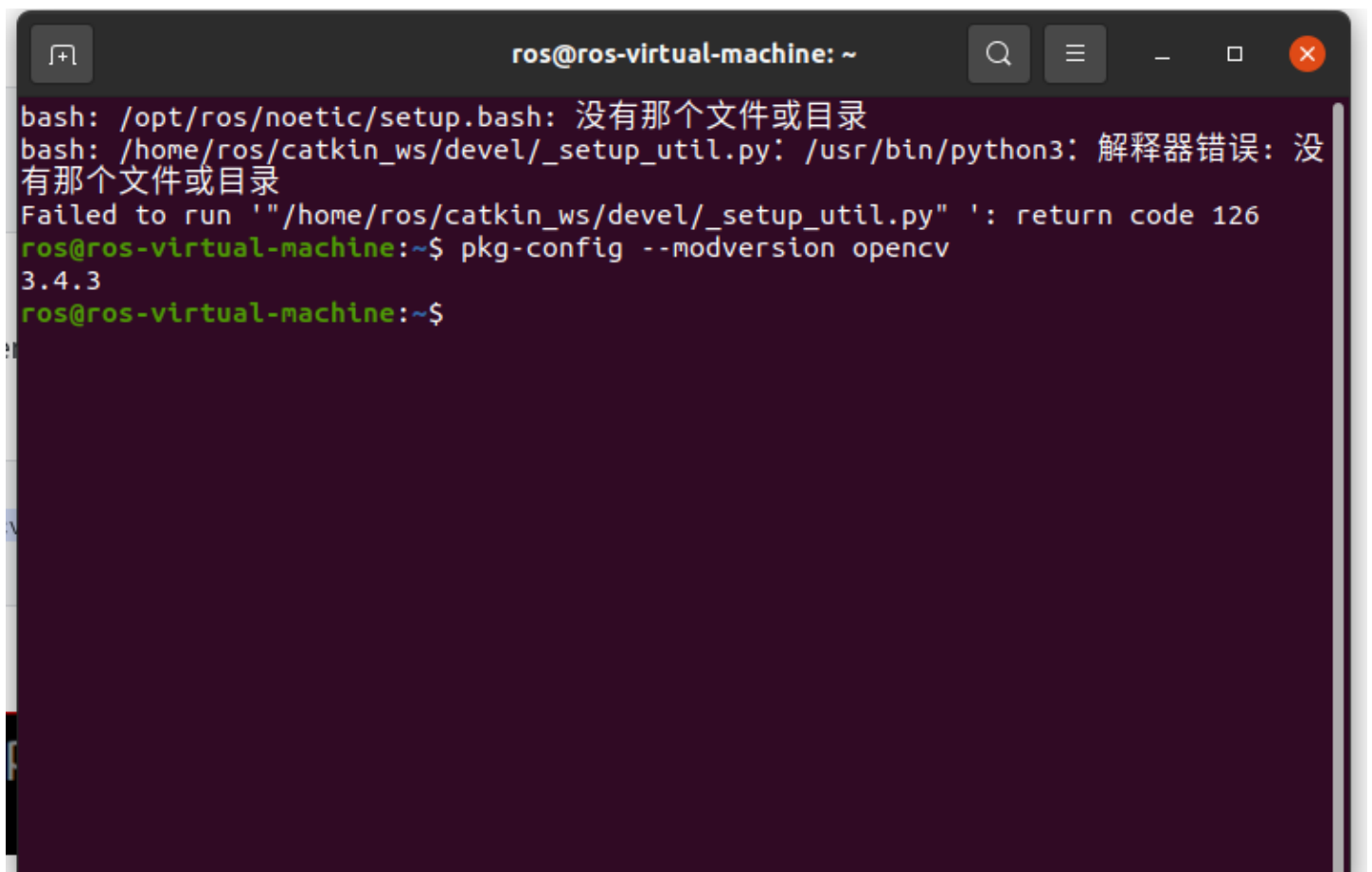
```
1 sudo updatedb
```

3.3.4.4 版本检测

安装好之后，我们肯定要对自己安装的OpenCV版本进行检测

输入下面的代码，查询OpenCV版本

```
1 pkg-config --modversion opencv
```

A terminal window titled 'ros@ros-virtual-machine: ~' with standard window controls. The terminal shows the following output:

```
bash: /opt/ros/noetic/setup.bash: 没有那个文件或目录
bash: /home/ros/catkin_ws/devel/_setup_util.py: /usr/bin/python3: 解释器错误: 没有那个文件或目录
Failed to run '"/home/ros/catkin_ws/devel/_setup_util.py" ': return code 126
ros@ros-virtual-machine:~$ pkg-config --modversion opencv
3.4.3
ros@ros-virtual-machine:~$
```

3.4 安装 boost 库

前往boost库官网进行下载，这里，笔者下载的是1.82.0版本

boost官网地址: <https://www.boost.org/>

VERSION 1.82.0

VERSION 1.82.0
April 14th, 2023 03:08 GMT
[Documentation](#)

DOWNLOADS

Platform	File	SHA256 Hash
unix	boost_1_82_0.tar.bz2	a6e1ab9b0860e6a2881dd7b21fe9f737a095e5f33a3a8743fc6a345228597ee6
	boost_1_82_0.tar.gz	66a469b6e608a51f8347236f4912e27dc5c60c60d7d53ae9bfe4683316c6f04c
windows	boost_1_82_0.7z	e72ff100482d4b371327b6c550dbd244d49c0186860ccc3b8c093dd621537e39
	boost_1_82_0.zip	ffc9e28d24abcd7a2c1b962039fdd463ca149d1883c3a950bbcc0ce6f7c6d9

KNOWN ISSUES

These are patches from library authors which were found too late to be fixed in the release.

- Filesystem
 - Compilation on OpenBSD may fail because of broken support for `-Wl, -no-undefined` compiler flag for shared libraries on this platform, see #283, Patch.
 - On Windows, creating directory iterators over directories in SMBv1 shares may fail with `ERROR_INVALID_LEVEL` error code, see #284, Patch.

NOTICE OF DROPPING C++03 SUPPORT

The following libraries will be dropping C++03 support in release 1.84, two releases from now:

Align
Any
Asio
Atomic
Bind
Chrono
ContainerHash
Conversion
DLL
Endian
Filesystem
Function
Functional
Io
LexicalCast
Log



Donate

VISA

WELCOME

INTRODUCTION

Download

News

Version History

License

People

Memoriam

Bibliography

Who's Using Boost

FAQ

How Boost Started

Index

COMMUNITY

DEVELOPMENT

DOCUMENTATION

下载之后，我们需要将下载的文件放在 homework_exercise 文件夹下并提取解压

解压后我们进入解压出来的文件夹，执行下面的指令

```
1 sudo ./bootstrap.sh
```

文件夹此时多了一些文件，我们再执行下面这个脚本

```
1 sudo ./b2 install
```

3.5 安装 libssl-dev

在 homework_exercise 文件夹下打开终端，输入下列命令行

```
1 sudo apt-get install libssl-dev
```

3.6 ORB-SLAM3 的编译和安装

3.6.1 ORB-SLAM3 编译

在 homework_exercise 文件夹下打开终端，输入下列命令行

```
1 cd ORB_SLAM3-master
2 chmod +x build.sh
```

打开ORB_SLAM3-master对应的CMakeLists.txt

找到 find_package(OpenCV 4.4)这行代码，将OpenCV版本号改为find_package(OpenCV 3.4)

3.6.2 安装 ORB-SLAM3

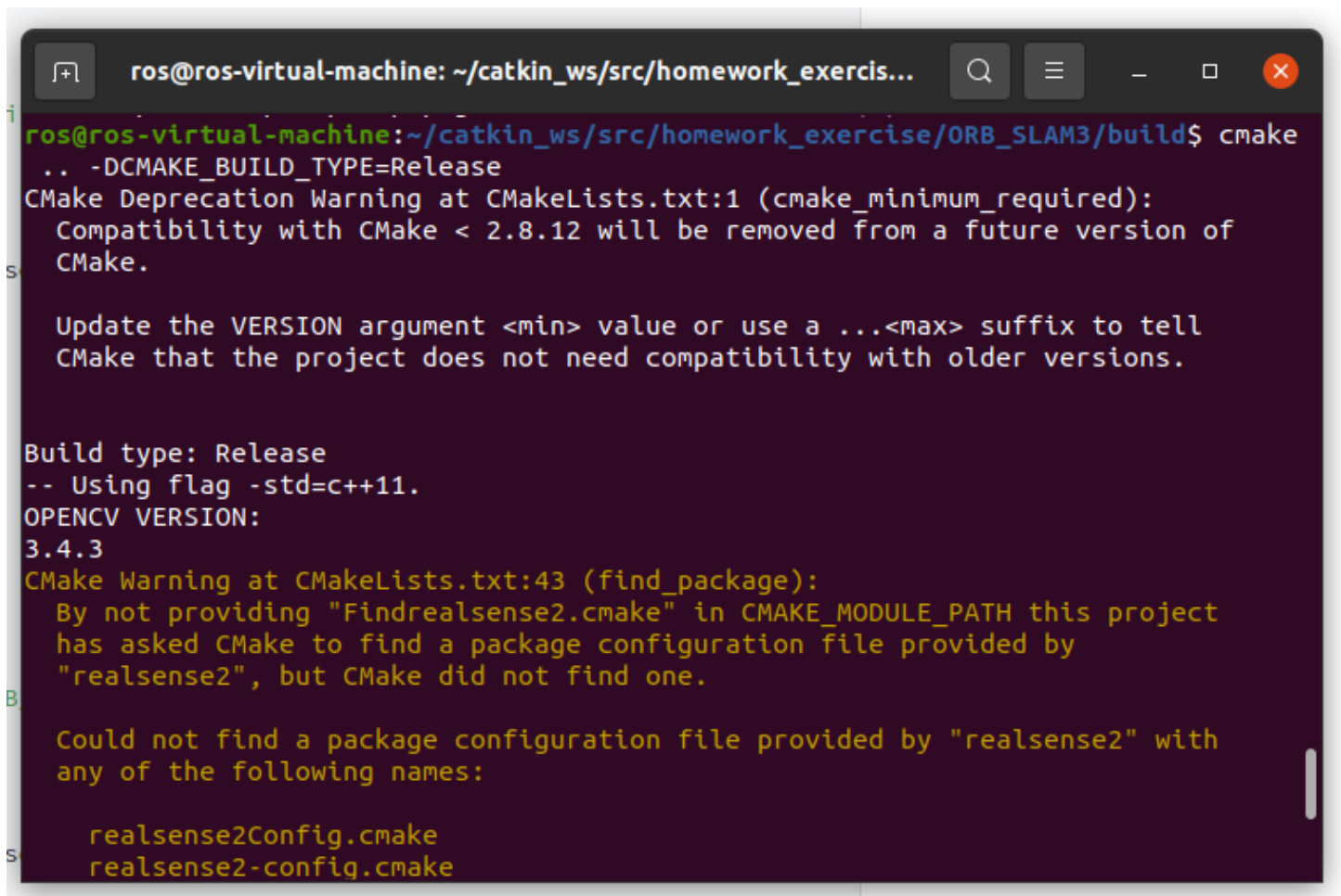
逐行依次执行下面的指令

```
1 echo "Configuring and building Thirdparty/DBow2 ..."
2
3 cd Thirdparty/DBow2
4 mkdir build
5 cd build
6 cmake .. -DCMAKE_BUILD_TYPE=Release
7 make -j
8
9 cd ../../g2o
10
11 echo "Configuring and building Thirdparty/g2o ..."
12
13 mkdir build
14 cd build
15 cmake .. -DCMAKE_BUILD_TYPE=Release
16 make -j
17
18 cd ../../Sophus
19
20 echo "Configuring and building Thirdparty/Sophus ..."
21
22 mkdir build
23 cd build
24 cmake .. -DCMAKE_BUILD_TYPE=Release
25 make -j
26
27 cd ../../../../
28
29 echo "Uncompress vocabulary ..."
30
31 cd Vocabulary
32 tar -xf ORBvoc.txt.tar.gz
33 cd ..
34
35 echo "Configuring and building ORB_SLAM3 ..."
```



```
36
37 mkdir build
38 cd build
39 cmake .. -DCMAKE_BUILD_TYPE=Release
40 make -j5
```

在执行到最后一个模块时，笔者报错

A terminal window titled 'ros@ros-virtual-machine: ~/catkin_ws/src/homework_exercis...' shows the execution of CMake. The prompt is 'ros@ros-virtual-machine:~/catkin_ws/src/homework_exercise/ORB_SLAM3/build\$'. The command 'cmake .. -DCMAKE_BUILD_TYPE=Release' is entered. The output includes a deprecation warning about CMake version compatibility, followed by build configuration details: 'Build type: Release', '-- Using flag -std=c++11.', and 'OPENCV VERSION: 3.4.3'. A warning at line 43 of CMakeLists.txt states that the project has asked CMake to find a package configuration file for 'realsense2' but it was not found. The error message lists the names CMake searched for: 'realsense2Config.cmake' and 'realsense2-config.cmake'.

```
ros@ros-virtual-machine:~/catkin_ws/src/homework_exercis...
ros@ros-virtual-machine:~/catkin_ws/src/homework_exercise/ORB_SLAM3/build$ cmake
.. -DCMAKE_BUILD_TYPE=Release
CMake Deprecation Warning at CMakeLists.txt:1 (cmake_minimum_required):
  Compatibility with CMake < 2.8.12 will be removed from a future version of
  CMake.

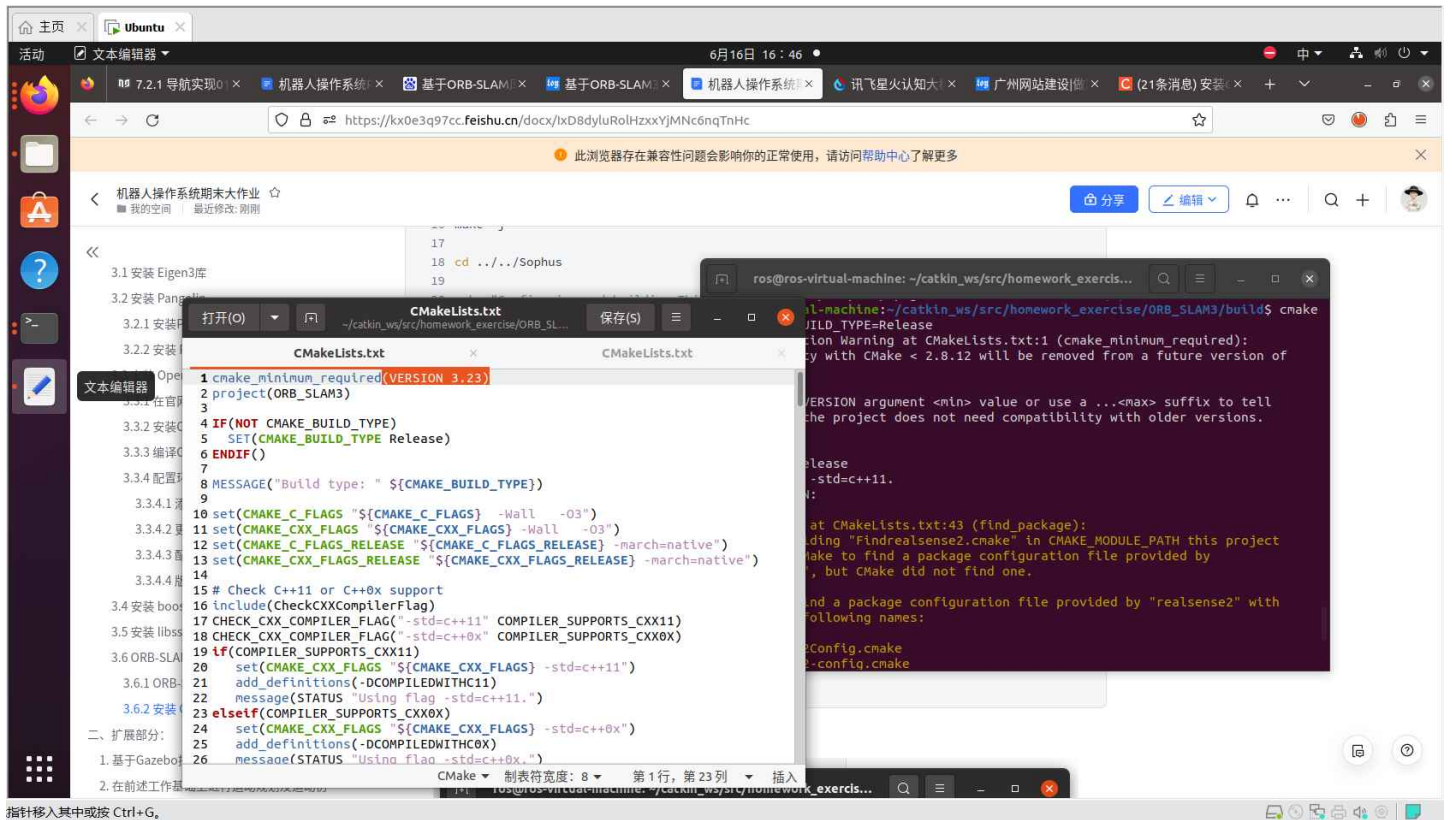
  Update the VERSION argument <min> value or use a ...<max> suffix to tell
  CMake that the project does not need compatibility with older versions.

Build type: Release
-- Using flag -std=c++11.
OPENCV VERSION:
3.4.3
CMake Warning at CMakeLists.txt:43 (find_package):
  By not providing "Findrealsense2.cmake" in CMAKE_MODULE_PATH this project
  has asked CMake to find a package configuration file provided by
  "realsense2", but CMake did not find one.

  Could not find a package configuration file provided by "realsense2" with
  any of the following names:

    realsense2Config.cmake
    realsense2-config.cmake
```

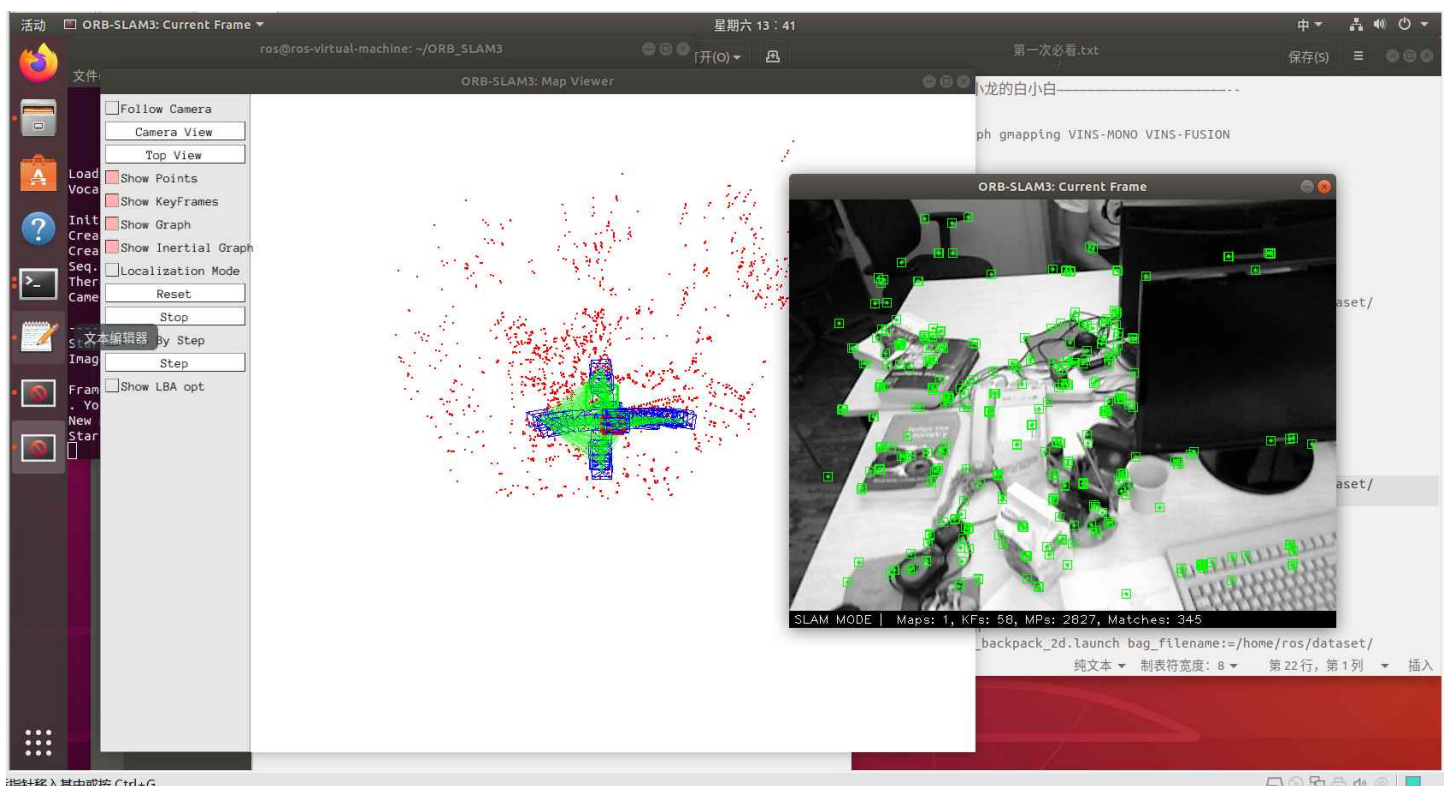
报错信息显示，我们需要将realsense2添加到路径中



4. 例程效果图

打开 ORB_SLAM3 文件夹，并输入以下指令，跑例程数据集

```
1 ./Examples/RGB-D/rgbd_tum Vocabulary/ORBvoc.txt Examples/RGB-D/TUM1.yaml
~/dataset/rgbd_dataset_freiburg1_xyz/
~/dataset/rgbd_dataset_freiburg1_xyz/associations.txt
```



二、扩展部分：

1. 基于Gazebo搭建移动机器人，并结合SLAM系统进行定位和建图仿真；

1.1 创建功能包，导入依赖项

创建新功能包，导入依赖包：urdf、xacro、gazebo_ros、gazebo_ros_control、gazebo_plugins

1.2 编写URDF或Xacro文件

1.2.1 编写封装惯性矩阵算法的xacro文件

```
1 <robot name="mycar" xmlns:xacro="http://wiki.ros.org/xacro">
2   <!--包含惯性矩阵文件-->
3   <xacro:include filename="head.xacro" />
4   <!--包含底盘、摄像头与雷达的xacro文件-->
5   <xacro:include filename="demo05_car_base.urdf.xacro" />
6   <xacro:include filename="demo06_car_camera.urdf.xacro" />
7   <xacro:include filename="demo07_car_laser.urdf.xacro" />
8   <!--运动控制-->
9   <xacro:include filename="gazebo/move.xacro" />
10  <!--laser lindar-->
11  <xacro:include filename="gazebo/laser.xacro" />
12  <!--camera-->
13  <xacro:include filename="gazebo/camera.xacro" />
14  <!--kinect-->
15  <xacro:include filename="gazebo/kinect.xacro" />
16
17 </robot>
18
19
```

搭建机器人模型

```
1 <!--
2   创建一个机器人模型(盒状即可)，显示在 Gazebo 中
3 -->
4
5 <robot name="mycar">
6   <link name="base_link">
7     <!--可视化部分-->
8     <visual>
9       <geometry>
```

```

10         <box size="0.5 0.2 0.1" />
11     </geometry>
12     <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
13     <material name="yellow">
14         <color rgba="0.5 0.3 0.0 1" />
15     </material>
16 </visual>
17
18 <!--设置碰撞参数-->
19 <collision>
20     <geometry>
21         <box size="0.5 0.2 0.1" />
22     </geometry>
23     <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
24 </collision>
25
26 <!--设置惯性矩阵-->
27 <inertial>
28     <origin xyz="0 0 0" />
29     <mass value="6" />
30     <inertia ixx="1" ixy="0" ixz="0" iyy="1" iyz="0" izz="1" />
31 </inertial>
32
33
34 </link>
35
36 <!--gazebo自己的颜色标签-->
37 <gazebo reference="base_link">
38     <material>Gazebo/Black</material>
39 </gazebo>
40
41 </robot>

```

1.2.2 编写模型相关xacro文件，并设置collision inertial以及color等相关参数

1.2.2.1 底盘Xacro文件

```

1 <!--
2     使用 xacro 优化 URDF 版的小车底盘实现：
3
4     实现思路：
5     1.将一些常量、变量封装为 xacro:property
6         比如:PI 值、小车底盘半径、离地间距、车轮半径、宽度 ....
7     2.使用 宏 封装驱动轮以及支撑轮实现，调用相关宏生成驱动轮与支撑轮
8

```

```
9 -->
10 <!-- 根标签，必须声明 xmlns:xacro -->
11 <robot name="my_base" xmlns:xacro="http://www.ros.org/wiki/xacro">
12     <!-- 封装变量、常量 -->
13     <!-- PI 值设置精度需要高一些，否则后续车轮翻转量计算时，可能会出现肉眼不能察觉的车轮倾
14     <xacro:property name="PI" value="3.1415926"/>
15     <!-- 宏:黑色设置 -->
16     <material name="black">
17         <color rgba="0.0 0.0 0.0 1.0" />
18     </material>
19     <!-- 底盘属性 -->
20     <xacro:property name="base_footprint_radius" value="0.001" /> <!-- base_foot
21     <xacro:property name="base_link_radius" value="0.1" /> <!-- base_link 半径 --
22     <xacro:property name="base_link_length" value="0.08" /> <!-- base_link 长 --
23     <xacro:property name="earth_space" value="0.015" /> <!-- 离地间距 -->
24     <xacro:property name="base_link_m" value="0.5" /> <!-- 质量 -->
25
26     <!-- 底盘 -->
27     <link name="base_footprint">
28         <visual>
29             <geometry>
30                 <sphere radius="${base_footprint_radius}" />
31             </geometry>
32         </visual>
33     </link>
34
35     <link name="base_link">
36         <visual>
37             <geometry>
38                 <cylinder radius="${base_link_radius}" length="${base_link_length}" />
39             </geometry>
40             <origin xyz="0 0 0" rpy="0 0 0" />
41             <material name="yellow">
42                 <color rgba="0.5 0.3 0.0 0.5" />
43             </material>
44         </visual>
45         <collision>
46             <geometry>
47                 <cylinder radius="${base_link_radius}" length="${base_link_length}" />
48             </geometry>
49             <origin xyz="0 0 0" rpy="0 0 0" />
50         </collision>
51         <xacro:cylinder_inertial_matrix m="${base_link_m}" r="${base_link_radius}"
52
53     </link>
54
55
```

```

56 <joint name="base_link2base_footprint" type="fixed">
57   <parent link="base_footprint" />
58   <child link="base_link" />
59   <origin xyz="0 0 ${earth_space + base_link_length / 2}" />
60 </joint>
61 <gazebo reference="base_link">
62   <material>Gazebo/Yellow</material>
63 </gazebo>
64
65 <!-- 驱动轮 -->
66 <!-- 驱动轮属性 -->
67 <xacro:property name="wheel_radius" value="0.0325" /><!-- 半径 -->
68 <xacro:property name="wheel_length" value="0.015" /><!-- 宽度 -->
69 <xacro:property name="wheel_m" value="0.05" /> <!-- 质量 -->
70
71 <!-- 驱动轮宏实现 -->
72 <xacro:macro name="add_wheels" params="name flag">
73   <link name="${name}_wheel">
74     <visual>
75       <geometry>
76         <cylinder radius="${wheel_radius}" length="${wheel_length}" />
77       </geometry>
78       <origin xyz="0.0 0.0 0.0" rpy="${PI / 2} 0.0 0.0" />
79       <material name="black" />
80     </visual>
81     <collision>
82       <geometry>
83         <cylinder radius="${wheel_radius}" length="${wheel_length}" />
84       </geometry>
85       <origin xyz="0.0 0.0 0.0" rpy="${PI / 2} 0.0 0.0" />
86     </collision>
87     <xacro:cylinder_inertial_matrix m="${wheel_m}" r="${wheel_radius}" h="${
88
89   </link>
90
91   <joint name="${name}_wheel2base_link" type="continuous">
92     <parent link="base_link" />
93     <child link="${name}_wheel" />
94     <origin xyz="0 ${flag * base_link_radius} ${-(earth_space + base_link_le
95     <axis xyz="0 1 0" />
96   </joint>
97
98   <gazebo reference="${name}_wheel">
99     <material>Gazebo/Red</material>
100   </gazebo>
101
102 </xacro:macro>

```

```

103 <xacro:add_wheels name="left" flag="1" />
104 <xacro:add_wheels name="right" flag="-1" />
105 <!-- 支撑轮 -->
106 <!-- 支撑轮属性 -->
107 <xacro:property name="support_wheel_radius" value="0.0075" /> <!-- 支撑轮半径
108 <xacro:property name="support_wheel_m" value="0.03" /> <!-- 质量 -->
109
110 <!-- 支撑轮宏 -->
111 <xacro:macro name="add_support_wheel" params="name flag" >
112   <link name="${name}_wheel">
113     <visual>
114       <geometry>
115         <sphere radius="${support_wheel_radius}" />
116       </geometry>
117       <origin xyz="0 0 0" rpy="0 0 0" />
118       <material name="black" />
119     </visual>
120     <collision>
121       <geometry>
122         <sphere radius="${support_wheel_radius}" />
123       </geometry>
124       <origin xyz="0 0 0" rpy="0 0 0" />
125     </collision>
126     <xacro:sphere_inertial_matrix m="${support_wheel_m}" r="${support_wheel_
127 </link>
128
129     <joint name="${name}_wheel2base_link" type="continuous">
130       <parent link="base_link" />
131       <child link="${name}_wheel" />
132       <origin xyz="${flag * (base_link_radius - support_wheel_radius)} 0 ${-
133       <axis xyz="1 1 1" />
134     </joint>
135     <gazebo reference="${name}_wheel">
136       <material>Gazebo/Red</material>
137     </gazebo>
138 </xacro:macro>
139
140 <xacro:add_support_wheel name="front" flag="1" />
141 <xacro:add_support_wheel name="back" flag="-1" />
142
143
144 </robot>
145

```

1.2.2.2 摄像头Xacro文件


```

1 <!-- 摄像头相关的 xacro 文件 -->
2 <robot name="my_camera" xmlns:xacro="http://wiki.ros.org/xacro">
3   <!-- 摄像头属性 -->
4   <xacro:property name="camera_length" value="0.01" /> <!-- 摄像头长度(x) -->
5   <xacro:property name="camera_width" value="0.025" /> <!-- 摄像头宽度(y) -->
6   <xacro:property name="camera_height" value="0.025" /> <!-- 摄像头高度(z) -->
7   <xacro:property name="camera_x" value="0.08" /> <!-- 摄像头安装的x坐标 -->
8   <xacro:property name="camera_y" value="0.0" /> <!-- 摄像头安装的y坐标 -->
9   <xacro:property name="camera_z" value="\${base_link_length} / 2 + camera_height" />
10
11   <xacro:property name="camera_m" value="0.01" /> <!-- 摄像头质量 -->
12
13   <!-- 摄像头关节以及link -->
14   <link name="camera">
15     <visual>
16       <geometry>
17         <box size="\${camera_length} \${camera_width} \${camera_height}" />
18       </geometry>
19       <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
20       <material name="black" />
21     </visual>
22     <collision>
23       <geometry>
24         <box size="\${camera_length} \${camera_width} \${camera_height}" />
25       </geometry>
26       <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
27     </collision>
28     <xacro:Box_inertial_matrix m="\${camera_m}" l="\${camera_length}" w="\${camera_width}" h="\${camera_height}" />
29   </link>
30
31   <joint name="camera2base_link" type="fixed">
32     <parent link="base_link" />
33     <child link="camera" />
34     <origin xyz="\${camera_x} \${camera_y} \${camera_z}" />
35   </joint>
36   <gazebo reference="camera">
37     <material>Gazebo/Blue</material>
38   </gazebo>
39 </robot>
40

```

1.2.2.3 雷达Xacro文件

```

1 <!--
2   小车底盘添加雷达

```



```

3 -->
4 <robot name="my_laser" xmlns:xacro="http://wiki.ros.org/xacro">
5
6     <!-- 雷达支架 -->
7     <xacro:property name="support_length" value="0.15" /> <!-- 支架长度 -->
8     <xacro:property name="support_radius" value="0.01" /> <!-- 支架半径 -->
9     <xacro:property name="support_x" value="0.0" /> <!-- 支架安装的x坐标 -->
10    <xacro:property name="support_y" value="0.0" /> <!-- 支架安装的y坐标 -->
11    <xacro:property name="support_z" value="\${base_link_length} / 2 + support_len
12
13    <xacro:property name="support_m" value="0.02" /> <!-- 支架质量 -->
14
15    <link name="support">
16        <visual>
17            <geometry>
18                <cylinder radius="\${support_radius}" length="\${support_length}"
19            </geometry>
20            <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
21            <material name="red">
22                <color rgba="0.8 0.2 0.0 0.8" />
23            </material>
24        </visual>
25
26        <collision>
27            <geometry>
28                <cylinder radius="\${support_radius}" length="\${support_length}"
29            </geometry>
30            <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
31        </collision>
32
33        <xacro:cylinder_inertial_matrix m="\${support_m}" r="\${support_radius}" h
34
35    </link>
36
37    <joint name="support2base_link" type="fixed">
38        <parent link="base_link" />
39        <child link="support" />
40        <origin xyz="\${support_x} \${support_y} \${support_z}" />
41    </joint>
42
43    <gazebo reference="support">
44        <material>Gazebo/White</material>
45    </gazebo>
46
47    <!-- 雷达属性 -->
48    <xacro:property name="laser_length" value="0.05" /> <!-- 雷达长度 -->
49    <xacro:property name="laser_radius" value="0.03" /> <!-- 雷达半径 -->

```

```

50 <xacro:property name="laser_x" value="0.0" /> <!-- 雷达安装的x坐标 -->
51 <xacro:property name="laser_y" value="0.0" /> <!-- 雷达安装的y坐标 -->
52 <xacro:property name="laser_z" value="${support_length / 2 + laser_length /
53
54 <xacro:property name="laser_m" value="0.1" /> <!-- 雷达质量 -->
55
56 <!-- 雷达关节以及link -->
57 <link name="laser">
58   <visual>
59     <geometry>
60       <cylinder radius="${laser_radius}" length="${laser_length}" />
61     </geometry>
62     <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
63     <material name="black" />
64   </visual>
65   <collision>
66     <geometry>
67       <cylinder radius="${laser_radius}" length="${laser_length}" />
68     </geometry>
69     <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
70   </collision>
71   <xacro:cylinder_inertial_matrix m="${laser_m}" r="${laser_radius}" h="${
72 </link>
73
74 <joint name="laser2support" type="fixed">
75   <parent link="support" />
76   <child link="laser" />
77   <origin xyz="${laser_x} ${laser_y} ${laser_z}" />
78 </joint>
79 <gazebo reference="laser">
80   <material>Gazebo/Black</material>
81 </gazebo>
82 </robot>
83

```

1.2.2.4 组合底盘、摄像头与雷达的Xacro文件

```

1 <!-- 组合小车底盘与摄像头 -->
2 <robot name="my_car_camera" xmlns:xacro="http://wiki.ros.org/xacro">
3   <xacro:include filename="my_head.urdf.xacro" />
4   <xacro:include filename="my_base.urdf.xacro" />
5   <xacro:include filename="my_camera.urdf.xacro" />
6   <xacro:include filename="my_laser.urdf.xacro" />

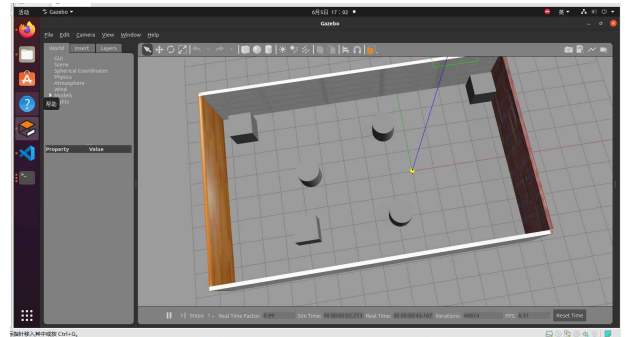
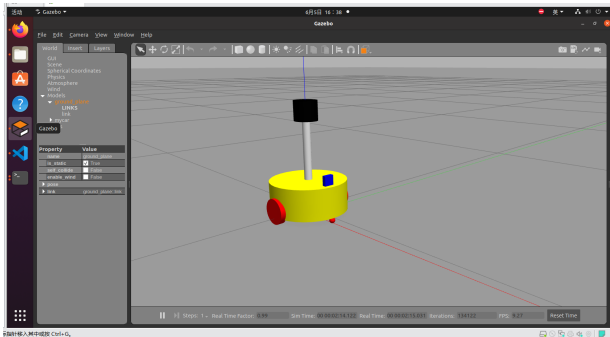
```

```
7 </robot>
8
```

1.3 启动Gazebo并显示机器人模型

编写launch文件实现

```
1 <launch>
2   <!-- 将 Urdf 文件的内容加载到参数服务器 -->
3   <param name="robot_description" command="$(find xacro)/xacro $(find demo02_u
4   <!-- 启动 gazebo -->
5   <include file="$(find gazebo_ros)/launch/empty_world.launch" />
6
7   <!-- 在 gazebo 中显示机器人模型 -->
8   <node pkg="gazebo_ros" type="spawn_model" name="model" args="-urdf -model my
9 </launch>
10
```



2. 在前述工作基础上进行运动规划及运动仿真。

2.1 运动控制及里程计信息显示

2.1.1 添加机器人传动装置

两轮差速配置

```
1 <robot name="my_car_move" xmlns:xacro="http://wiki.ros.org/xacro">
2
3   <!-- 传动实现:用于连接控制器与关节 -->
4   <xacro:macro name="joint_trans" params="joint_name">
5     <!-- Transmission is important to link the joints and the controller -->
```

```

6      <transmission name="${joint_name}_trans">
7          <type>transmission_interface/SimpleTransmission</type>
8          <joint name="${joint_name}">
9              <hardwareInterface>hardware_interface/VelocityJointInterface</ha
10         </joint>
11         <actuator name="${joint_name}_motor">
12             <hardwareInterface>hardware_interface/VelocityJointInterface</ha
13             <mechanicalReduction>1</mechanicalReduction>
14         </actuator>
15     </transmission>
16 </xacro:macro>
17
18 <!-- 每一个驱动轮都需要配置传动装置 -->
19 <xacro:joint_trans joint_name="left_wheel2base_link" />
20 <xacro:joint_trans joint_name="right_wheel2base_link" />
21
22 <!-- 控制器 -->
23 <gazebo>
24     <plugin name="differential_drive_controller" filename="libgazebo_ros_dif
25         <rosDebugLevel>Debug</rosDebugLevel>
26         <publishWheelTF>true</publishWheelTF>
27         <robotNamespace>/</robotNamespace>
28         <publishTf>1</publishTf>
29         <publishWheelJointState>true</publishWheelJointState>
30         <alwaysOn>true</alwaysOn>
31         <updateRate>100.0</updateRate>
32         <legacyMode>true</legacyMode>
33         <leftJoint>left_wheel2base_link</leftJoint> <!-- 左轮 -->
34         <rightJoint>right_wheel2base_link</rightJoint> <!-- 右轮 -->
35         <wheelSeparation>${base_link_radius * 2}</wheelSeparation> <!-- 车轮i
36         <wheelDiameter>${wheel_radius * 2}</wheelDiameter> <!-- 车轮直径 -->
37         <broadcastTF>1</broadcastTF>
38         <wheelTorque>30</wheelTorque>
39         <wheelAcceleration>1.8</wheelAcceleration>
40         <commandTopic>cmd_vel</commandTopic> <!-- 运动控制话题 -->
41         <odometryFrame>odom</odometryFrame>
42         <odometryTopic>odom</odometryTopic> <!-- 里程计话题 -->
43         <robotBaseFrame>base_footprint</robotBaseFrame> <!-- 根坐标系 -->
44     </plugin>
45 </gazebo>
46
47 </robot>
48

```

2.1.2 xacro文件集成

```

1 <!-- 组合小车底盘与摄像头 -->
2 <robot name="my_car_camera" xmlns:xacro="http://wiki.ros.org/xacro">
3   <xacro:include filename="my_head.urdf.xacro" />
4   <xacro:include filename="my_base.urdf.xacro" />
5   <xacro:include filename="my_camera.urdf.xacro" />
6   <xacro:include filename="my_laser.urdf.xacro" />
7   <xacro:include filename="move.urdf.xacro" />
8 </robot>
9

```

注意，但前核心包含控制器以及传动配置的xacro文件

```

1 <xacro:include filename="move.urdf.xacro" />

```

2.1.3 启动gazebo控制机器人运动

```

1 <launch>
2
3   <!-- 将 Urdf 文件的内容加载到参数服务器 -->
4   <param name="robot_description" command="$(find xacro)/xacro $(find demo02_u
5   <!-- 启动 gazebo -->
6   <include file="$(find gazebo_ros)/launch/empty_world.launch">
7     <arg name="world_name" value="$(find demo02_urdf_gazebo)/worlds/hello.wo
8   </include>
9
10  <!-- 在 gazebo 中显示机器人模型 -->
11  <node pkg="gazebo_ros" type="spawn_model" name="model" args="-urdf -model my
12 </launch>
13

```

输入下面的命令行，可以使用命令控制

```

1 rostopic pub -r 10 /cmd_vel geometry_msgs/Twist '{linear: {x: 0.2, y: 0, z: 0},

```

2.1.4 Rviz查看里程计信息

```

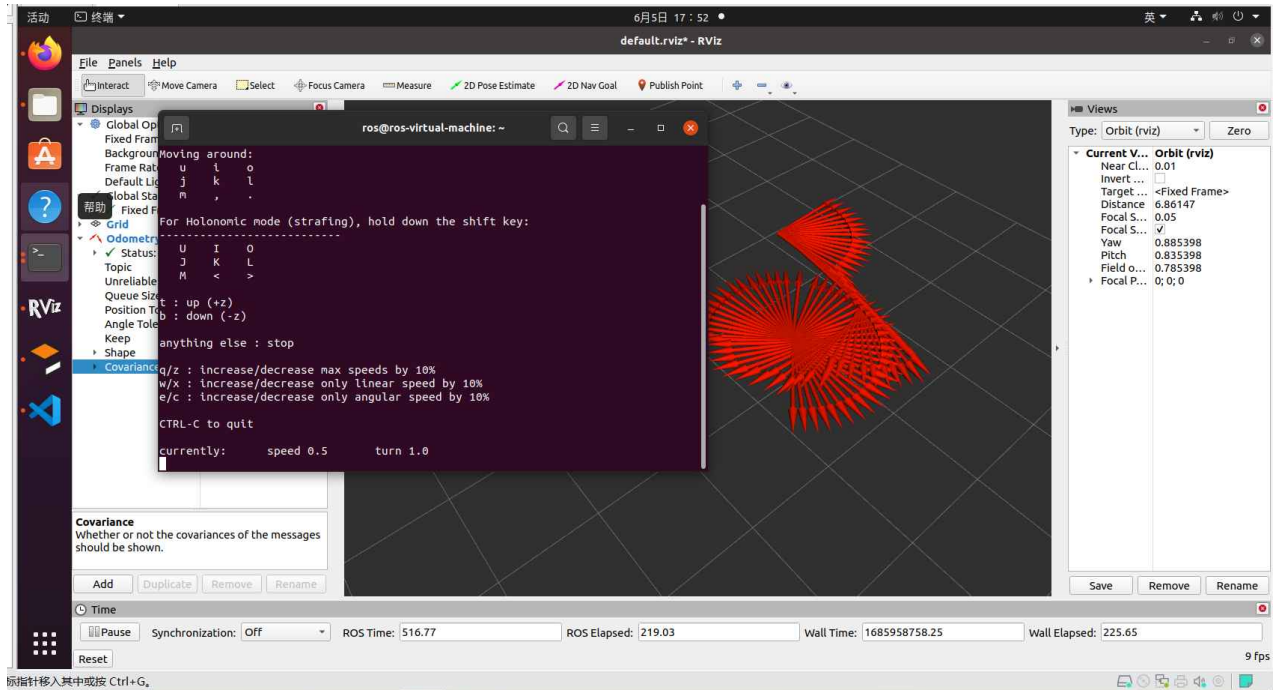
1 <launch>
2   <!-- 启动 rviz -->

```

```

3   <node pkg="rviz" type="rviz" name="rviz" />
4
5   <!-- 关节以及机器人状态发布节点 -->
6   <node name="joint_state_publisher" pkg="joint_state_publisher" type="joint_s
7   <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_s
8
9 </launch>
10

```



2.2 雷达信息仿真以及显示

2.2.1 新建xacro文件，配置雷达传感器信息

```

1 <robot name="my_sensors" xmlns:xacro="http://wiki.ros.org/xacro">
2
3   <!-- 雷达 -->
4   <gazebo reference="laser">
5     <sensor type="ray" name="rplidar">
6       <pose>0 0 0 0 0 0</pose>
7       <visualize>true</visualize>
8       <update_rate>5.5</update_rate>
9     </ray>
10    <scan>
11      <horizontal>
12        <samples>360</samples>

```

```

13         <resolution>1</resolution>
14         <min_angle>-3</min_angle>
15         <max_angle>3</max_angle>
16     </horizontal>
17 </scan>
18 <range>
19     <min>0.10</min>
20     <max>30.0</max>
21     <resolution>0.01</resolution>
22 </range>
23 <noise>
24     <type>gaussian</type>
25     <mean>0.0</mean>
26     <stddev>0.01</stddev>
27 </noise>
28 </ray>
29 <plugin name="gazebo_rplidar" filename="libgazebo_ros_laser.so">
30     <topicName>/scan</topicName>
31     <frameName>laser</frameName>
32 </plugin>
33 </sensor>
34 </gazebo>
35
36 </robot>
37

```

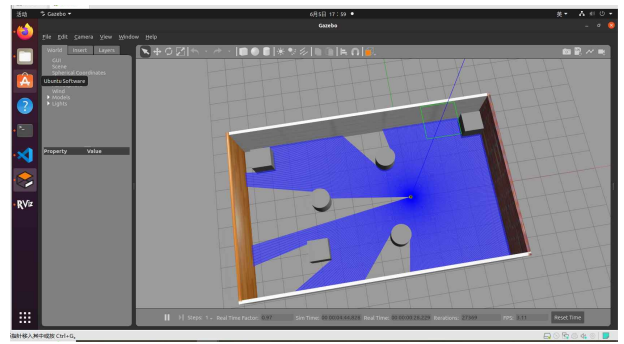
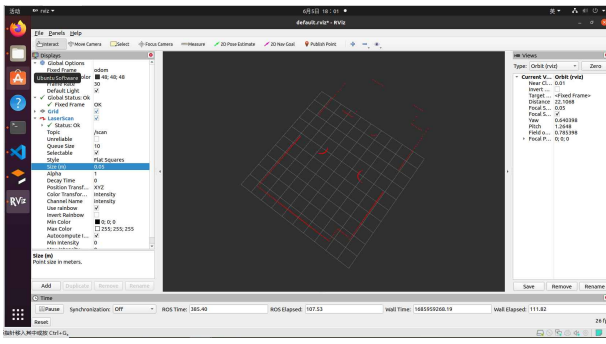
2.2.2 xacro文件集成

```

1  <!-- 组合小车底盘与传感器 -->
2  <robot name="my_car_camera" xmlns:xacro="http://wiki.ros.org/xacro">
3      <xacro:include filename="my_head.urdf.xacro" />
4      <xacro:include filename="my_base.urdf.xacro" />
5      <xacro:include filename="my_camera.urdf.xacro" />
6      <xacro:include filename="my_laser.urdf.xacro" />
7      <xacro:include filename="move.urdf.xacro" />
8      <!-- 雷达仿真的 xacro 文件 -->
9      <xacro:include filename="my_sensors_laser.urdf.xacro" />
10 </robot>
11

```

2.2.3 编写launch文件，启动gazebo仿真环境



2.3 摄像头信息仿真及显示

2.3.1 新建xacro文件，配置摄像头传感器信息

```

1 <robot name="my_sensors" xmlns:xacro="http://wiki.ros.org/xacro">
2   <!-- 被引用的link -->
3   <gazebo reference="camera">
4     <!-- 类型设置为 camera -->
5     <sensor type="camera" name="camera_node">
6       <update_rate>30.0</update_rate> <!-- 更新频率 -->
7       <!-- 摄像头基本信息设置 -->
8       <camera name="head">
9         <horizontal_fov>1.3962634</horizontal_fov>
10        <image>
11          <width>1280</width>
12          <height>720</height>
13          <format>R8G8B8</format>
14        </image>
15        <clip>
16          <near>0.02</near>
17          <far>300</far>
18        </clip>
19        <noise>
20          <type>gaussian</type>
21          <mean>0.0</mean>
22          <stddev>0.007</stddev>
23        </noise>
24      </camera>
25    <!-- 核心插件 -->
26    <plugin name="gazebo_camera" filename="libgazebo_ros_camera.so">
27      <alwaysOn>true</alwaysOn>
28      <updateRate>0.0</updateRate>
29      <cameraName>/camera</cameraName>
30      <imageTopicName>image_raw</imageTopicName>
31      <cameraInfoTopicName>camera_info</cameraInfoTopicName>
32      <frameName>camera</frameName>

```



```

33     <hackBaseline>0.07</hackBaseline>
34     <distortionK1>0.0</distortionK1>
35     <distortionK2>0.0</distortionK2>
36     <distortionK3>0.0</distortionK3>
37     <distortionT1>0.0</distortionT1>
38     <distortionT2>0.0</distortionT2>
39   </plugin>
40 </sensor>
41 </gazebo>
42 </robot>
43

```

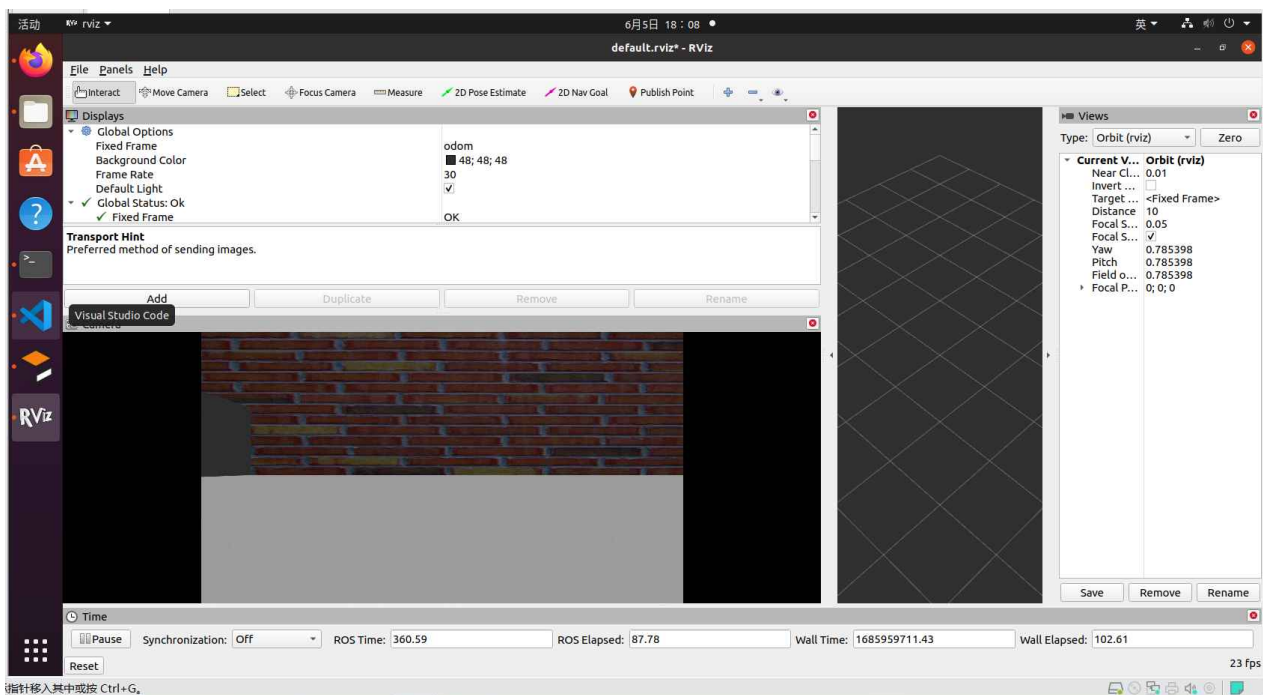
2.3.2 xacro文件集成

```

1  <!-- 组合小车底盘与传感器 -->
2  <robot name="my_car_camera" xmlns:xacro="http://wiki.ros.org/xacro">
3    <xacro:include filename="my_head.urdf.xacro" />
4    <xacro:include filename="my_base.urdf.xacro" />
5    <xacro:include filename="my_camera.urdf.xacro" />
6    <xacro:include filename="my_laser.urdf.xacro" />
7    <xacro:include filename="move.urdf.xacro" />
8    <!-- 摄像头仿真的 xacro 文件 -->
9    <xacro:include filename="my_sensors_camara.urdf.xacro" />
10 </robot>
11

```

2.3.3 编写launch文件，启动gazebo仿真环境



2.4 kinect信息仿真及显示

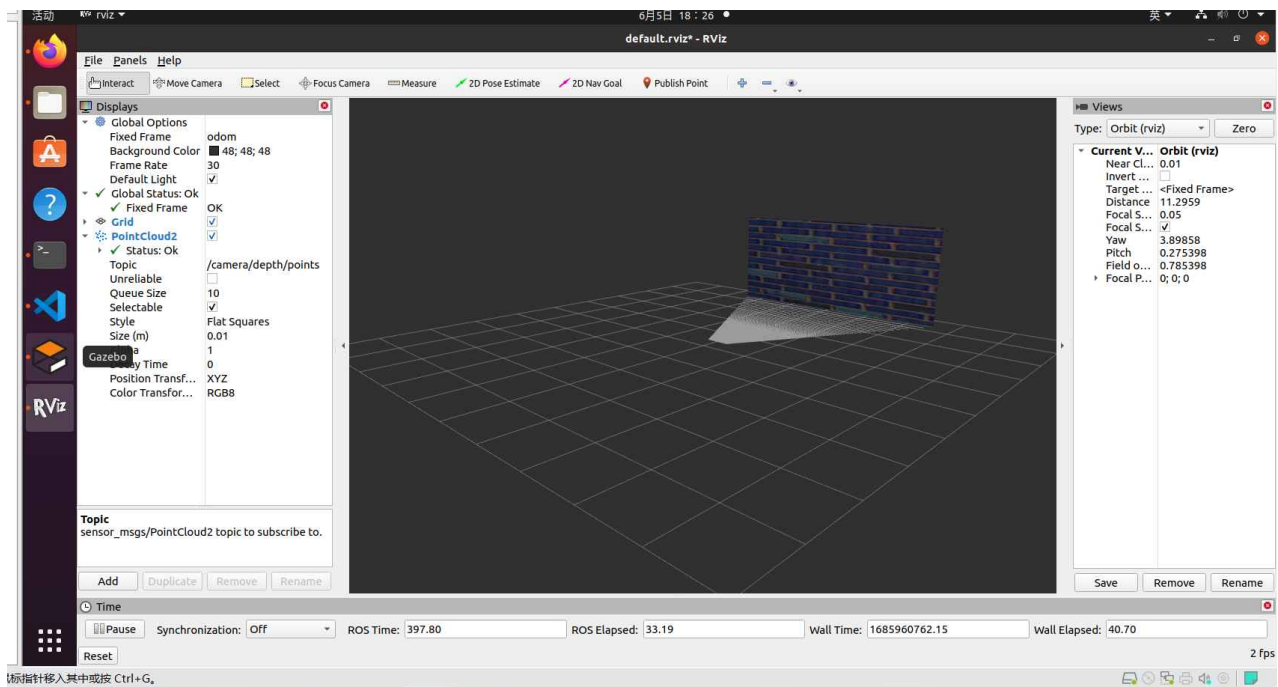
2.4.1 新建xacro文件，配置kinetic传感器信息

```
1 <robot name="my_sensors" xmlns:xacro="http://wiki.ros.org/xacro">
2   <gazebo reference="kinect link名称">
3     <sensor type="depth" name="camera">
4       <always_on>true</always_on>
5       <update_rate>20.0</update_rate>
6       <camera>
7         <horizontal_fov>${60.0*PI/180.0}</horizontal_fov>
8         <image>
9           <format>R8G8B8</format>
10          <width>640</width>
11          <height>480</height>
12        </image>
13        <clip>
14          <near>0.05</near>
15          <far>8.0</far>
16        </clip>
17      </camera>
18      <plugin name="kinect_camera_controller" filename="libgazebo_ros_openni_k
19        <cameraName>camera</cameraName>
20        <alwaysOn>true</alwaysOn>
21        <updateRate>10</updateRate>
22        <imageTopicName>rgb/image_raw</imageTopicName>
23        <depthImageTopicName>depth/image_raw</depthImageTopicName>
24        <pointCloudTopicName>depth/points</pointCloudTopicName>
25        <cameraInfoTopicName>rgb/camera_info</cameraInfoTopicName>
26        <depthImageCameraInfoTopicName>depth/camera_info</depthImageCameraInfo
27        <frameName>kinect link名称</frameName>
28        <baseline>0.1</baseline>
29        <distortion_k1>0.0</distortion_k1>
30        <distortion_k2>0.0</distortion_k2>
31        <distortion_k3>0.0</distortion_k3>
32        <distortion_t1>0.0</distortion_t1>
33        <distortion_t2>0.0</distortion_t2>
34        <pointCloudCutoff>0.4</pointCloudCutoff>
35      </plugin>
36    </sensor>
37  </gazebo>
38
39 </robot>
40
```

2.4.2 xacro文件集成

```
1 <!-- 组合小车底盘与传感器 -->
2 <robot name="my_car_camera" xmlns:xacro="http://wiki.ros.org/xacro">
3   <xacro:include filename="my_head.urdf.xacro" />
4   <xacro:include filename="my_base.urdf.xacro" />
5   <xacro:include filename="my_camera.urdf.xacro" />
6   <xacro:include filename="my_laser.urdf.xacro" />
7   <xacro:include filename="move.urdf.xacro" />
8   <!-- kinect仿真的 xacro 文件 -->
9   <xacro:include filename="my_sensors_kinect.urdf.xacro" />
10 </robot>
11
```

2.4.3 编写launch文件，启动gazebo仿真环境



三、附录

1. git安装

```
1 sudo apt-get install git
```

2. vi编辑器安装

```
1 # 有些新系统的vi编辑器不完整，需要重新安装
2 sudo apt-get remove vim-common
3 sudo apt-get install vim
```

四、Q&A

这个部分是笔者自己做得时候遇到的一些通用问题，其余过程中遇到的问题文档相应位置，其他资料可参考本文第五部分“参考资料”。



- 遇到问题好心态：不要怕，总能解决的，早晚的事情
- 解决问题的思路：查看报错信息 -> 在网站上搜索相关解决办法 -> 并不是所有问题都能够找到，但是能够找到相似的解决方法 -> 根据网上的提示解决问题

1. git clone

Q1: unable to access 'https://github.com/'

这个问题可能是git配置的问题，也有可能是网络的问题，最暴力的解决办法是登陆具体的网址，直接下载代码文件，所以，

- 解决方案一：最直接的办法，直接登陆具体的网址，下载代码文件
- 解决方案二：查看git配置

输入下面的指令行，如果没有任何与https代理相关的内容，则没有问题

```
1 git config --global -l
```

如果有相关内容，请输入下列指令，并删除对应的内容，然后重试

```
1 ~/ .gitconfig
```

- 解决方案三：手动配置git的代理

git客户端输入如下两个命令就可以了。

```
1 git config --global http.proxy http://127.0.0.1:1080
2 git config --global https.proxy http://127.0.0.1:1080
```

2. command

Q1:Command ‘make ‘ not found, but can be installed...

这个指令是说没有找到对应指令集，当然也有可能是路径的问题，我们可以通过重新安装解决这个问题

比如我们可以依次执行下列指令

```
1 sudo apt-cdrom add
2 sudo apt-get update
3 sudo apt-get install build-essential
```

Q2: catkin_make:Command ‘catkin_make ‘ not found, but can be installed with:sudo apt install catkin

同理上面，我们只需要依次执行下列指令行即可

```
1 source /opt/ros/melodic/setup.bash
2 echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
3 source ~/.bashrc
```

Q3: Linux环境下出现 cmake: command not found

这里，我们针对环境变量中缺少路径的情况提出解决办法

暂时添加环境变量关闭窗口后就失效了，不推荐，这里我们推荐永久添加环境变量（如下）

```
1 $ cd
2 $ touch .bash_profile
3 $ vim .bash_profile
4 export CMAKE_ROOT=/Applications/CMake.app/Contents/bin/
5 export PATH=$CMAKE_ROOT:$PATH
6 $ source .bash_profile
```

添加后，我们可以开一个新窗口，查看cmake版本号

```
1 $ cmake --version
```

五、参考资料

1. 基于ORB-SLAM3库搭建SLAM系统
2. [git clone出现 fatal: unable to access 'https://github.com/...'的解决办法\(亲测有效\)](#)
3. [catkin_make:Command 'catkin_make' not found, but can be installed with:sudo apt install catkin_Mr_D0](#)
4. [报错Command 'make' not found, but can be installed..._command make_Ham235XueBi的博客-CSDN博客](#)
5. [Mac/Linux环境下出现 cmake: command not found_songarpore的博客-CSDN博客](#)
6. [BUG List_Hangro的博客-CSDN博客](#)（这篇文档记录了超级全的bug指南，推荐）