

# ORB\_SLAM2 安装与运行(Linux 18.04下测试)

---

## 依赖项

### Pangolin

github地址:<https://github.com/stevenlovegrove/Pangolin.git>

安装方式

```
1  # Required
2  # OpenGL (Desktop / ES / ES2)
3  sudo apt install libgl1-mesa-dev
4  # Glew
5  sudo apt install libglew-dev
6  # CMake
7  sudo apt install cmake
8
9  # Recommended
10 # Python2 / Python3, for drop-down interactive console
11 sudo apt install libpython2.7-dev
12 # Wayland
13 sudo apt install pkg-config
14 sudo apt install libegl1-mesa-dev libwayland-dev libxkbcommon-dev wayland-
   protocols
15
16 # Optional
17 # FFMPEG (For video decoding and image rescaling)
18 sudo apt install ffmpeg libavcodec-dev libavutil-dev libavformat-dev libswscale-dev
   libavdevice-dev
19 # DC1394 (For firewire input)
20 sudo apt install libdc1394-22-dev libraw1394-dev
21 # libuvc (For cross-platform webcam video input via libusb)
22 git://github.com/ktossell/libuvc.git
23 # libjpeg, libpng, libtiff, libopenexr (For reading still-image sequences)
24 sudo apt install libjpeg-dev libpng-dev libtiff5-dev libopenexr-dev
25
```

```
26 # 以上依赖除了Required项，其他的可以自己选装，对ORB_SLAM2的编译影响不大
27
28 # build and install
29 git clone https://github.com/stevenlovegrove/Pangolin.git
30 cd Pangolin
31 mkdir build
32 cd build
33 cmake ..
34 make
35 sudo make install
```

## Opencv

需要Opencv版本 $\geq 2.4.11$ ，安装方式可以参见[Opencv安装文档](#)

## Eigen3

Eigen3版本需要 $\geq 3.1.0$ ，一般来说16.04和18.04使用apt安装的版本就符合这个要求

```
1 sudo apt-get install libeigen3-dev
```

## DBow2 and g2o

已经包含在ORB\_SLAM2中的第三方库文件夹中，不需要自己编译安装

## ROS

ROS的安装方式参见[ROS melodic install help](#)。此外，对于目前的SLAM领域来说，不求深刻掌握，但了解ROS的基本使用方式，大致看的懂ROS的代码是非常有必要的。对于学习ROS，最推荐的还是官方的21讲，里面将ROS的基本工作原理和使用方式讲的非常清楚，具体参见[ROS Toturials](#)，看 [Beginner Level](#) 就可以了。

## ORB\_SLAM库文件编译(包含基本例程)

```
1 git clone https://github.com/raulmur/ORB_SLAM2.git ORB_SLAM2
2 cd ORB_SLAM2
3 chmod +x build.sh
4 ./build.sh
```

其中遇到的最主要的问题为

```
1 error: 'usleep' was not declared in this scope
```

解决方式为在 `ORB_SLAM2/include/System.h` 中添加一句

```
1 #include "unistd.h"
```

运行结束后将会生成 `libORB_SLAM2.so` 在 `lib` 文件夹中 以及测试程序 `mono_tum`, `mono_kitti`, `rgbd_tum`, `stereo_kitti`, `mono_euroc` 和 `stereo_euroc` 在 `Example` 文件夹中。

## 单目示例

此处以单目模式下的TUM数据集为例，演示以下作者提供的例程如何运行。

首先从 <http://vision.in.tum.de/data/datasets/rgbd-dataset/download> 中下载任意一个数据包并解压，这里以 `fr1/xyz` 为例。解压后的路径为 `/home/teamo/rgbd_dataset_freiburg1_xyz`

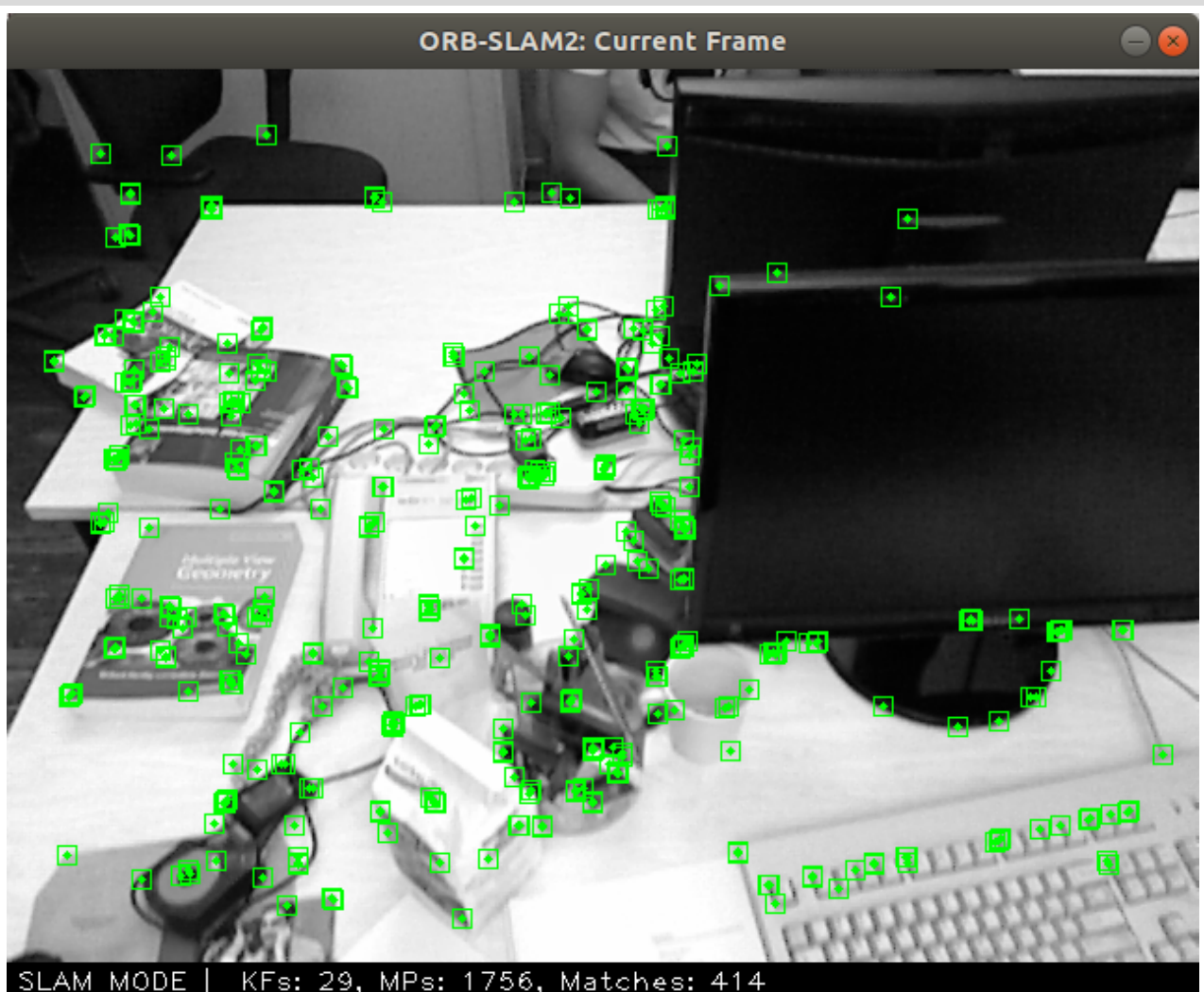
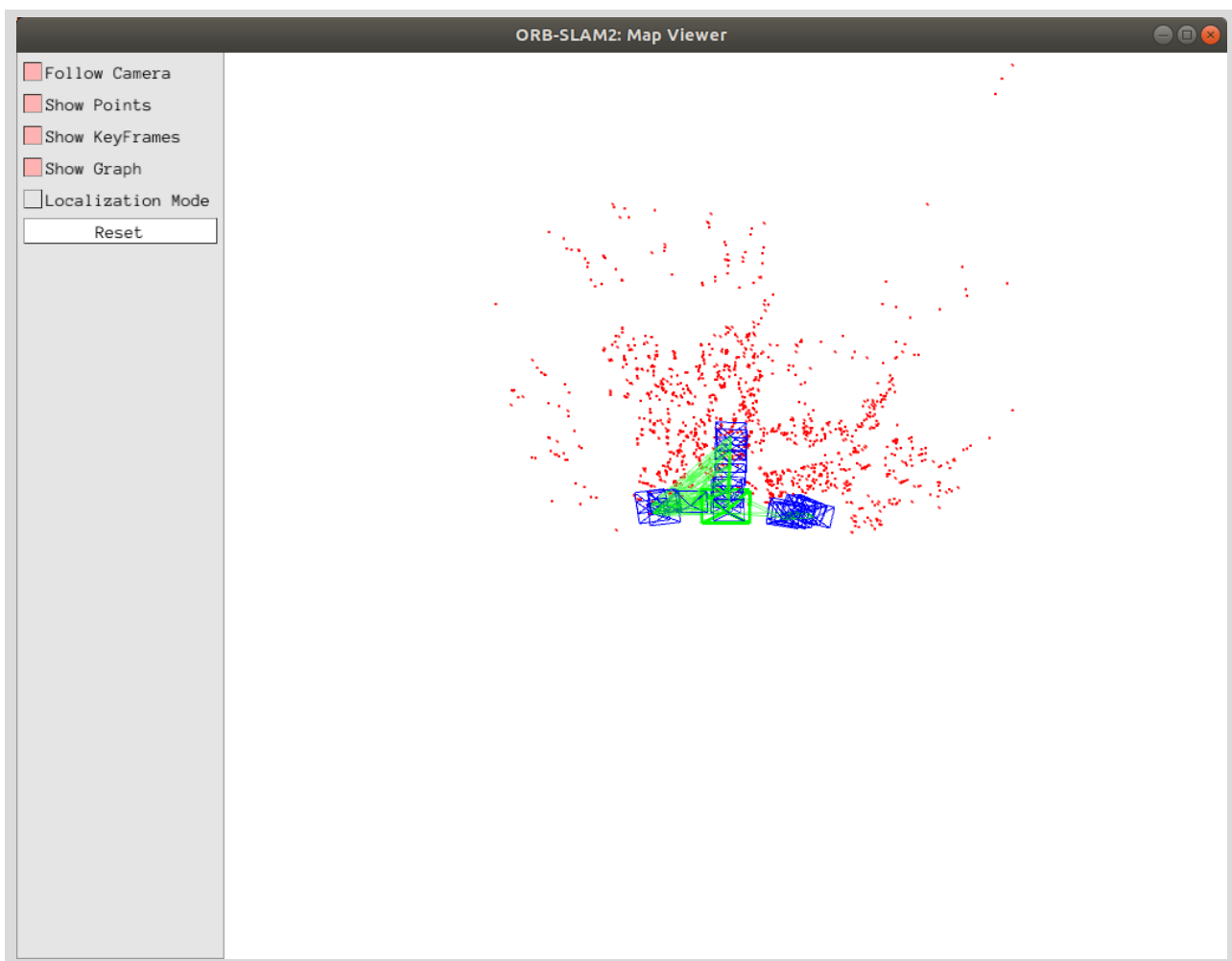
接下来执行以下命令

```
1 ./Examples/Monocular/mono_tum Vocabulary/ORBvoc.txt
  Examples/Monocular/TUMX.yaml PATH_TO_SEQUENCE_FOLDER
```

注意，此处的 `TUMX.yaml` 文件要对应于你下载的数据集类型，`PATH_TO_SEQUENCE_FOLDER` 要对应于你的数据集文件夹路径，在我的电脑上，上述命令应修改如下：

```
1 ./Examples/Monocular/mono_tum Vocabulary/ORBvoc.txt
  Examples/Monocular/TUM1.yaml /home/teamo/rgbd_dataset_freiburg1_xyz/
```

运行结果如下



## ROS示例

### 构建

首先需要将 `Examples/ROS/ORB_SLAM2` 路径导入到 `ROS_PACKAGE_PATH` 环境变量中，具体做法为打开终端，输入 `sudo gedit ~/.bashrc`，并在文件最后一行添加

```
1 export
  ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:PATH/ORB_SLAM2/Examples/ROS
```

此处的 `PATH` 为你的 `ORB_SLAM2` 文件夹所在的绝对路径，比如在我的电脑上需要添加

```
1 export
  ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:/home/teamo/ORB_SLAM2/Examples
  /ROS
```

之后执行构建脚本

```
1 chmod +x build_ros.sh
2 ./build_ros.sh
```

如果仍然出现以下错误

```
1 [rosbuild] rospack found package "ORB_SLAM2" at "", but the current
2 directory is "/home/teamo/ORB_SLAM2/Examples/ROS/ORB_SLAM2". You should
3 double-check your ROS_PACKAGE_PATH to ensure that packages are found in the
4 correct precedence order.
```

则需要 `在 /opt/ros/<你的ros版本>/setup.bash 文件中最后一行添加`

```
1 export
  ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:/home/teamo/ORB_SLAM2/Examples
  /ROS
```

之后重新执行构建脚本即可。

若在编译过程中出现如下错误

```
1 /usr/bin/ld: CMakeFiles/RGBD.dir/src/ros_rgbd.cc.o: undefined reference to symbol  
'_ZN5boost6system15system_categoryEv'
```

则需要修改ORB\_SLAM2/Examples/ROS/ORB\_SLAM2C中的CMakeLists.txt文件，将第51行的代码修改如下

```
1 set(LIBS  
2   ${OpenCV_LIBS}  
3   ${EIGEN3_LIBS}  
4   ${Pangolin_LIBRARIES}  
5   ${PROJECT_SOURCE_DIR}/../Thirdparty/DBoW2/lib/libDBoW2.so  
6   ${PROJECT_SOURCE_DIR}/../Thirdparty/g2o/lib/libg2o.so  
7   ${PROJECT_SOURCE_DIR}/../lib/libORB_SLAM2.so  
8   -lboost_system)
```

之后重新构建即可。

## 例程

以双目模式为例，演示ROS下的ORB\_SLAM2运行结果。

首先，从<https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdataset>s上下载一个roslab文件，我这里选择的是Machine Hall 01，解压之后的放置路径为/home/teamo/ORB\_SLAM2/MH\_01\_easy.bag

之后打开三个不同的终端，运行以下三句代码

```
1 roscore
```

```
1 roslaunch ORB_SLAM2 Stereo Vocabulary/ORBvoc.txt Examples/Stereo/EuRoC.yaml true
```

```
1 roslaunch play --pause /home/teamo/ORB_SLAM2/MH_01_easy.bag  
/cam0/image_raw:=/camera/left/image_raw  
/cam1/image_raw:=/camera/right/image_raw
```

这里的后两句为ROS的消息映射，因为bag包中的消息与作者程序中接收的消息名称并不一致。

需要注意的是，我们需要等待ORB词典读取完成后，再切换到播放bag的shell并按下空格开始播放，若提前播放，会丢失部分数据。

其实，仔细分析ros代码后可以发现，与非ros的例程并没有本质区别，只是将按文件名逐帧读取图片更改为了按照固定频率接受ros消息后读取图片，其他调用方式与非ros基本一样。

结果如下

