ORB-SLAM3安装与运行

记录编译安装 `ORB_SLAM3` ,并运行EuRoC 数据集的ROS bag

首先，默认 `g++`, `cmake`， `make`, `ROS` 等等最基本工具已经安装。

ROS 安装可以参考 blog 或其他教程

依赖库安装

Eigen 3.3.4

下载 `Eigen3.3.4` 并进行编译与安装 （目前编译ORB_SLAM3 时 `Eigen` 这个版本 (>3.3.0) 没遇到问题，版本过高或过低可能会遇到一些问题，需要修改CMakeList 等等，编译ORB_SLAM2 需要Eigen 3.2）

注意尽量不要用 `apt` 来装eigen，很可能找不到或版本不兼容，可以 `sudo apt remove libeigen3-dev` 将之前这样装的eigen 卸载。

下载所需版本的源码： 发布 · libeigen / eigen · GitLab

```
tar -xzvf eigen-3.3.4.tar.gz
//如果下载的是.zip:
unzip eigen-3.3.4.zip

cd eigen-3.3.4
mkdir build && cd build
cmake ..
sudo make install
```

查看自己下载的Eigen 版本方法：

`cat /usr/include/eigen3/Eigen/src/Core/util/Macros.h | grep VERSION` 或

`cat /usr/local/include/eigen3/Eigen/src/Core/util/Macros.h | grep VERSION`

Pangolin 0.6

Pangolin 0.6 下载源码

按照首页readme安装GitHub - stevenlovegrove/Pangolin at v0.6

```
sudo apt install libgl1-mesa-dev
sudo apt install libglew-dev
sudo apt install libpython2.7-dev
sudo apt install pkg-config
sudo apt install libegl1-mesa-dev libwayland-dev libxkbcommon-dev wayland-protocols


unzip Pangolin-0.6.zip
cd Pangolin
mkdir build
cd build
cmake ..
make -j3
sudo make install
```

测试：

```
cd Pangolin-0.6/examples/HelloPangolin
mkdir build
cd build
cmake ..
make
./HelloPangolin
```

成功运行一个红绿蓝立方体

Boost 库安装

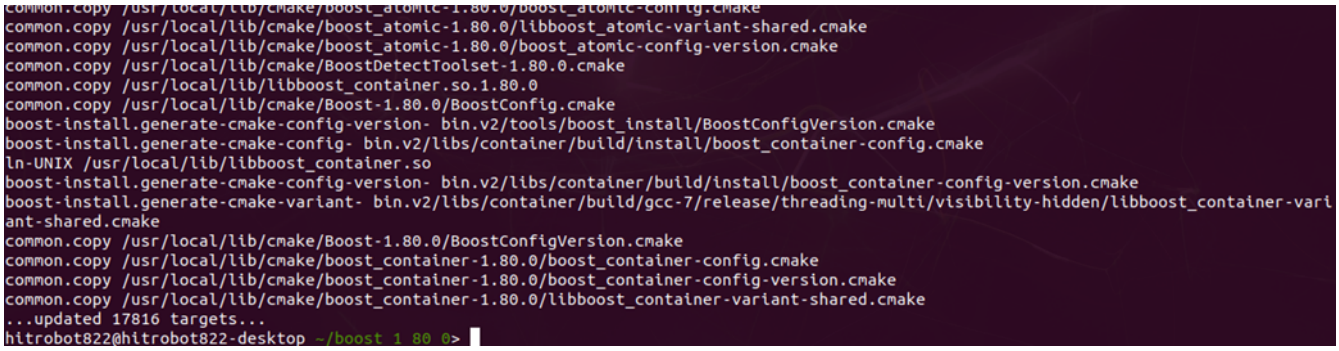
boost 1.80.0 下载源码

解压到合适位置

```
cd boost-1.80.0
sudo ./bootstrap.sh   //这一步较占内存可能卡死
```

`bootstrap.sh` 会生成 `b2` 工具，继续执行

```
sudo ./b2 install
```



之后 `/usr/local/include` 下会有boost的头文件，`/usr/local/lib` 下面会生成boost库

下载编译ORB-SLAM 3 源码

ORB-SLAM3

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

```
cd ORB_SLAM3
cat build.sh
```

一步一步在命令行手动执行 `build. sh` 中的命令，这样可以方便解决报错

```
echo "Configuring and building Thirdparty/DBoW2 ..."

cd Thirdparty/DBoW2
mkdir build
cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
make -j

cd ../../g2o

echo "Configuring and building Thirdparty/g2o ..."
```

```
mkdir build
cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
make -j

cd ../../Sophus

echo "Configuring and building Thirdparty/Sophus ..."

mkdir build
cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
make -j

cd ../../../

echo "Uncompress vocabulary ..."

cd Vocabulary
tar -xf ORBvoc.txt.tar.gz
cd ..

echo "Configuring and building ORB_SLAM3 ..."

mkdir build
cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
make -j
```

先编译安装 `Thirdparty` 目录下的 `DBoW2`，`g2o`，`Sophus`，再解压 `Vocabulary` 下词典，最后编译 `ORB-SLAM`，在 `lib/` 下生成 `libORB_SLAM3.so`，并编译好 `Example/` 下的执行程序。

我当时只编译安装 `Thirdparty` 目录下的 `DBoW2`，`g2o`，`Sophus`，再解压 `Vocabulary` 下词典，因为要通过ROS运行，没有进行最后一步编译 `ORB-SLAM`，在 `lib/` 下生成 `libORB_SLAM3.so`。（后面又回来执行了最后一步）

同样一步步执行 `build_ros.sh` 的内容：

```
cd Examples/ROS/ORB_SLAM3
mkdir build
cd build
cmake .. -DROS_BUILD_TYPE=Release
make -j
```

但最新的官方源码里的 `Examples` 下没有ROS 目录，可能是误删了，从 `Examples_old/` 下复制或者从这个详细注释版本 ORB_SLAM3_detailed_comments 里相同目录下复制过去即可。

编译时会报错：

```
-- Found PythonInterp: /usr/bin/python (found version "2.7.17")
[rosbuild] Building package ORB_SLAM3
[rosbuild] Error from directory check: /opt/ros/melodic/share/ros/core/rosbuild/bin/check_same_d
irectories.py  /home/hitrobot822/Desktop/ORB_SLAM3/Examples_old/ROS/ORB_SLAM3
1
Traceback (most recent call last):
  File "/opt/ros/melodic/share/ros/core/rosbuild/bin/check_same_directories.py", line 46, in <mo
dule>
    raise Exception
Exception
CMake Error at /opt/ros/melodic/share/ros/core/rosbuild/private.cmake:99 (message):
  [rosbuild] rospack found package "ORB_SLAM3" at "", but the current
  directory is
  "/home/hitrobot822/Desktop/ORB_SLAM3/Examples_old/ROS/ORB_SLAM3".  You
  should double-check your ROS_PACKAGE_PATH to ensure that packages are found
  in the correct precedence order.
Call Stack (most recent call first):
  /opt/ros/melodic/share/ros/core/rosbuild/public.cmake:177 (_rosbuild_check_package_location)
  CMakeLists.txt:4 (rosbuild_init)


-- Configuring incomplete, errors occurred!
See also "/home/hitrobot822/Desktop/ORB_SLAM3/Examples_old/ROS/ORB_SLAM3/build/CMakeFiles/CMakeO
utput.log".
```

没有添加该目录为 `ROS_PACKAGE_PATH`

修改 `~/.bashrc` ，最后加一句：`export ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:/home/hitrobot822/Desktop/ORB_SLAM3/Examples/ROS`

重新运行 `cmake .. -DROS_BUILD_TYPE=Release`

报错：

```
CMake Warning at CMakeLists.txt:35 (find_package):
  Could not find a configuration file for package "OpenCV" that is compatible
  with requested version "3.2".

  The following configuration files were considered but not accepted:

    /usr/lib/aarch64-linux-gnu/cmake/opencv4/OpenCVConfig.cmake, version: 4.1.1


CMake Error at CMakeLists.txt:37 (message):
  OpenCV > 3.2 not found.
```

是 `opencv` 出现问题

`opencv` 安装

我根据这个博客： [ORBSLAM3 安装及测试教程（Ubuntu20.04） - 滑稽果 - 博客园](#)安装了opencv。

opencv 下载 4.4.0 源码 （其实最好是安装 3.4.0 版本的，但我因为先安装了 4.4.0，3.4.0 出现问题装不上，可以参照 ubuntu18.04系统,opencv3.4.9+contrib完全安装指南 博客尝试安装 3.4 版本的 `opencv`）

opencv contrib 下载与**opencv 版本一致**的扩展库

目录结构为

```
opencv
| - opencv-4.4.0      // 4.4.0 源码
| - opencv_contrib-4.4.0  //与opencv 版本一致的扩展库
```

https://wwtt.lanzouw.com/if60o1cwvv4h 密码: d5fx 下载 `boostdesc_bgm` 与 `vgg_generated`，将这里面的十几个文件放到 `opencv_contrib-4.4.0/modules/xfeatures2d/src` 目录中

修改 `opencv_contrib-4.4.0/modules/xfeatures2d/test/test_features2d.cpp`

第 51~52 行代码路径改为

```
#include "../../../../opencv-4.4.0/modules/features2d/test/test_detectors_regression.impl.hpp"
#include "../../../../opencv-4.4.0/modules/features2d/test/test_descriptors_regression.impl.hpp"
```

修改 `opencv_contrib-4.4.0/modules/xfeatures2d/test/test_rotation_and_scale_invariance.cpp` 文件，将里面的第 7~8 行代码路径改为：

```
#include "../../../../opencv-4.4.0/modules/features2d/test/test_detectors_invariance.impl.hpp" // main OpenCV repo
#include "../../../../opencv-4.4.0/modules/features2d/test/test_descriptors_invariance.impl.hpp" // main OpenCV repo
```

然后编译安装

```
cd opencv
mkdir -p build && cd build
cmake -DOPENCV_EXTRA_MODULES_PATH= ../opencv_contrib-4.4.0/modules   ../opencv-4.4.0
make -j4    #这步就要开四个，编译很慢
sudo make install # 别忘了
```

`opencv` 在 `make` 过程中报错：

```
[ 53%] Built target opencv_gapi
[ 53%] Building CXX object modules/ximgproc/CMakeFiles/opencv_ximgproc.dir/src/guided_filter.cpp.o
Consolidate compiler generated dependencies of target opencv_sfm
[ 53%] Building CXX object modules/sfm/CMakeFiles/opencv_sfm.dir/src/simple_pipeline.cpp.o
/home/hitrobot822/opencv/opencv_contrib-4.4.0/modules/sfm/src/simple_pipeline.cpp:44:10: fatal error: opencv2
/xfeatures2d.hpp: No such file or directory
 #include <opencv2/xfeatures2d.hpp>
          ^~~~~~~~~~~~~~~~~~~~~~~~~
compilation terminated.
modules/sfm/CMakeFiles/opencv_sfm.dir/build.make:173: recipe for target 'modules/sfm/CMakeFiles/opencv_sfm.di
r/src/simple_pipeline.cpp.o' failed
make[2]: *** [modules/sfm/CMakeFiles/opencv_sfm.dir/src/simple_pipeline.cpp.o] Error 1
CMakeFiles/Makefile2:7268: recipe for target 'modules/sfm/CMakeFiles/opencv_sfm.dir/all' failed
make[1]: *** [modules/sfm/CMakeFiles/opencv_sfm.dir/all] Error 2
make[1]: *** Waiting for unfinished jobs....
[ 53%] Building CXX object modules/ximgproc/CMakeFiles/opencv_ximgproc.dir/src/joint_bilateral_filter.cpp.o
```

依照 blog 解决

将 `opencv_contrib-4.4.0\modules\xfeatures2d \include\opencv2` 此目录下所有文件复制到opencv的安装位置 `opencv\build\include\opencv2` 中，这样 `#include<opencv2/xfeatures2.hpp>` 不会报错

继续 `make -j4` 成功编译

```
Consolidate compiler generated dependencies of target opencv_stereo
[100%] Built target opencv_stereo
[100%] Building CXX object modules/python2/CMakeFiles/opencv_python2.dir/__/src2/cv2.cpp.o
[100%] Linking CXX executable ../../bin/opencv_perf_optflow
[100%] Built target opencv_perf_optflow
[100%] Building CXX object modules/python3/CMakeFiles/opencv_python3.dir/__/src2/cv2.cpp.o
[100%] Linking CXX executable ../../bin/opencv_perf_superres
[100%] Built target opencv_perf_superres
[100%] Building CXX object modules/stereo/CMakeFiles/opencv_perf_stereo.dir/perf/perf_bm.cpp.o
[100%] Building CXX object modules/stereo/CMakeFiles/opencv_test_stereo.dir/test/test_block_matching.cpp.o
[100%] Building CXX object modules/stereo/CMakeFiles/opencv_test_stereo.dir/test/test_descriptors.cpp.o
[100%] Building CXX object modules/stereo/CMakeFiles/opencv_perf_stereo.dir/perf/perf_descriptor.cpp.o
[100%] Building CXX object modules/stereo/CMakeFiles/opencv_test_stereo.dir/test/test_main.cpp.o
[100%] Linking CXX executable ../../bin/opencv_test_stereo
[100%] Built target opencv_test_stereo
[100%] Building CXX object modules/stereo/CMakeFiles/opencv_perf_stereo.dir/perf/perf_main.cpp.o
[100%] Linking CXX executable ../../bin/opencv_perf_stereo
[100%] Built target opencv_perf_stereo
[100%] Linking CXX shared module ../../lib/python3/cv2.cpython-36m-aarch64-linux-gnu.so
[100%] Linking CXX shared module ../../lib/cv2.so
[100%] Built target opencv_python3
[100%] Built target opencv_python2
hitrobot822@hitrobot822-desktop ~/o/build> ls
```

继续编译ORB-SLAM 3

重新编译 `Examples/ROS/ORB_SLAM3/`， 根据 `build_ros.sh` 的内容：

```
cd Examples/ROS/ORB_SLAM3
mkdir build
cd build
cmake .. -DROS_BUILD_TYPE=Release
make -j
```

然而 `cmake .. -DROS_BUILD_TYPE=Release` 依旧会报一样的错

应该是opencv 4.0 太新了，但上面又说大于 3.2 即可，有点凌乱，又有博客说 4.4 没问题。

最后直接用一种粗暴方式解决了：

直接修改 `CMakeList.txt`, 将

```
find_package(OpenCV 3.0 QUIET)
if(NOT OpenCV_FOUND)
   find_package(OpenCV 2.4.3 QUIET)
   if(NOT OpenCV_FOUND)
      message(FATAL_ERROR "OpenCV > 2.4.3 not found.")
   endif()
endif()
```

第一行修改为 `find_package(OpenCV 4.4)`

重新 `make`

提示缺少 `libORB_SLAM3.so`

回去在 `ORB_SLAM3/build` 下

`cmake .. -DCMAKE_BUILD_TYPE=Release`

`make -j` （很慢，极容易卡死）

成功编译

```
[ 75%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/hyper_graph_action.cpp.o
[ 76%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/hyper_graph.cpp.o
[ 77%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/marginal_covariance_cholesky.cpp.o
[ 78%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/matrix_structure.cpp.o
[ 79%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/batch_stats.cpp.o
[ 80%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/parameter.cpp.o
[ 81%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/cache.cpp.o
[ 82%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/optimizable_graph.cpp.o
[ 82%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/solver.cpp.o
[ 83%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/optimization_algorithm_factory.cpp.o
[ 84%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/estimate_propagator.cpp.o
[ 85%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/factory.cpp.o
[ 86%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/sparse_optimizer.cpp.o
[ 87%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/hyper_dijkstra.cpp.o
[ 88%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/parameter_container.cpp.o
[ 89%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/optimization_algorithm.cpp.o
[ 90%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/optimization_algorithm_with_hessian.cpp.o
[ 91%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/optimization_algorithm_levenberg.cpp.o
[ 92%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/optimization_algorithm_gauss_newton.cpp.o
[ 93%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/jacobian_workspace.cpp.o
[ 94%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/robust_kernel.cpp.o
[ 95%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/robust_kernel_factory.cpp.o
[ 95%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/core/robust_kernel_impl.cpp.o
[ 96%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/stuff/timeutil.cpp.o
[ 97%] Building C object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/stuff/os_specific.c.o
[ 98%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/stuff/string_tools.cpp.o
[ 99%] Building CXX object Thirdparty/g2o/CMakeFiles/g2o.dir/g2o/stuff/property.cpp.o
[100%] Linking CXX shared library ../../../Thirdparty/g2o/lib/libg2o.so
[100%] Built target g2o
hitrobot822@hitrobot822-desktop ~/D/O/build>
```

接着 `Examples/ROS/ORB_SLAM3/build` 下执行编译完，也通过了！

运行EuRoC双目数据集

从 kmavvisualinertialdatasets – ASL Datasets 下载Euroc 双目数据集

Dataset	ROS bag	ASL Dataset Format	Comment
Machine Hall 01	link	link	Dataset machine hall “easy”
Machine Hall 02	link	link	Dataset machine hall “easy”
Machine Hall 03	link	link	Dataset machine hall “medium”
Machine Hall 04	link	link	Dataset machine hall “difficult”
Machine Hall 05	link	link	Dataset machine hall “difficult”
Vicon Room 1 01	link	link	Dataset Vicon room 1 “easy”
Vicon Room 1 02	link	link	Dataset Vicon room 1 “medium”
Vicon Room 1 03	link	link	Dataset Vicon room 1 “difficult”
Vicon Room 2 01	link	link	Dataset Vicon room 2 “easy”
Vicon Room 2 02	link	link	Dataset Vicon room 2 “medium”
Vicon Room 2 03	link	link	Dataset Vicon room 2 “difficult”
Calibration Dataset	link	link	Dataset for custom calibration

我选择了MH 03 的ROS bag，但官网下的很卡而且ubuntu 上可能下不了，最好搜索搜索在百度网盘 SLAM数据集（百度网盘）_ntu rgb d 120深度 数据集下载百度网盘-CSDN博客 上下到自己电脑上，然后u 盘拷到板子上。

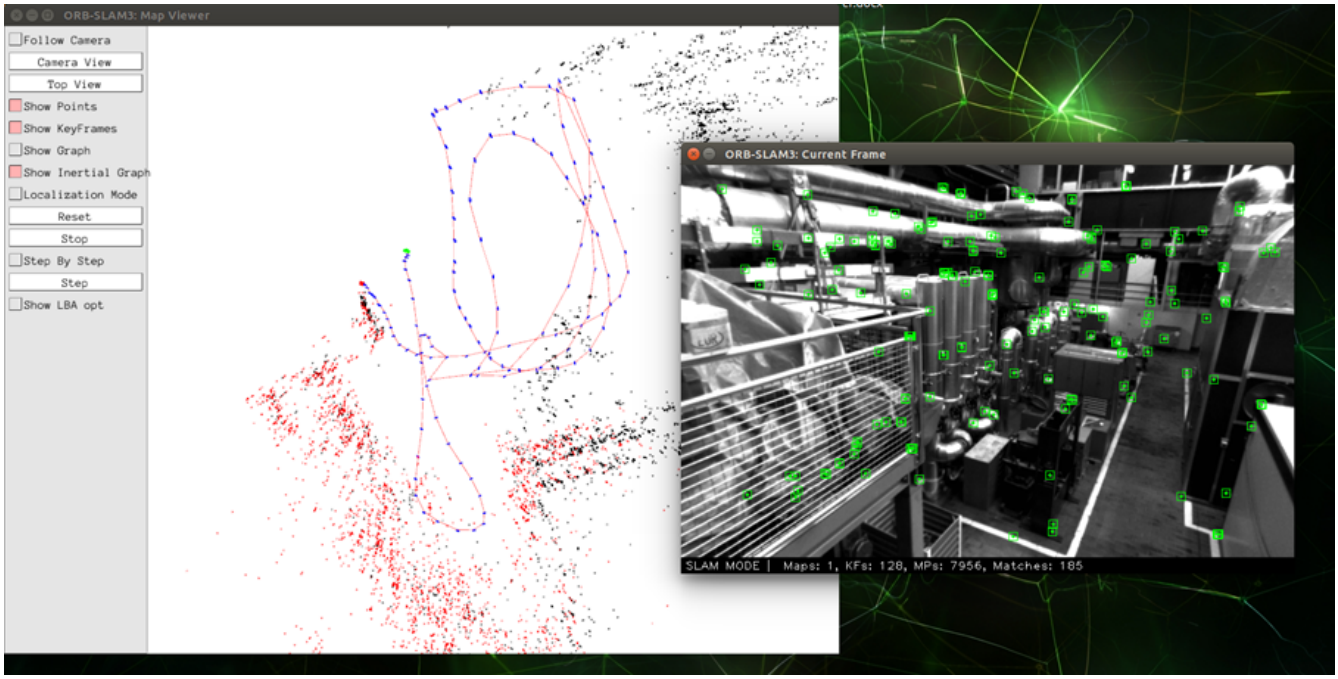
打开一个终端运行 `roscore`

另开一个运行 `rosrun ORB_SLAM3 Stereo_Inertial Vocabulary/ORBvoc.txt Examples/Stereo-Inertial/EuRoC.yaml false`

（为什么这里是false，参见 https://github.com/UZ-SLAMLab/ORB_SLAM3/issues/237，不显式打出默认就是false，显式用true 会运行失败）

再开一个运行bag：`rosbag play dataset/MH_03_medium.bag /cam0/image_raw:=/camera/left/image_raw /cam1/image_raw:=/camera/right/image_raw /imu0:=/imu`

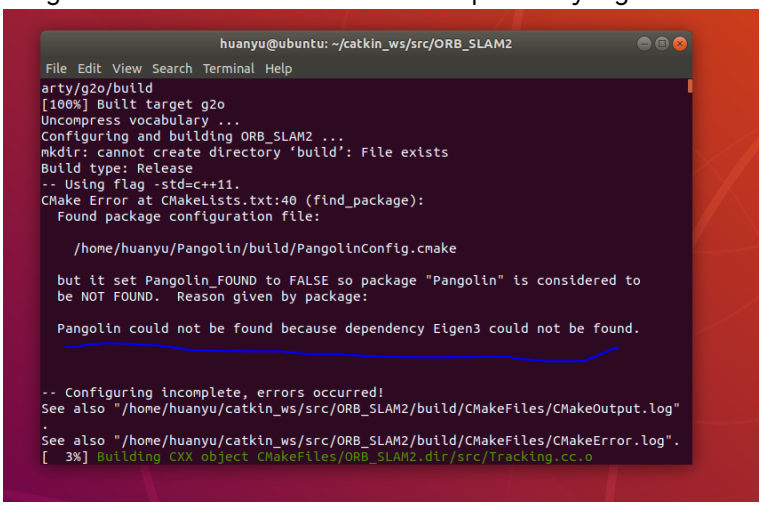
成功运行：



一些可能遇到的问题

在虚拟机编译ORB_SLAM 2可能遇到的问题

1. Pangolin could not be found because dependency Eigen 3 could not be found



解决：[Pangolin could not be found because dependency Eigen3 could not be found · Issue #1015 · raulmur/ORB_SLAM2 · GitHub]

2. 编译ORB_SLAM2报错 'usleep' was not declared in this scope：

```
/home/sankuai/Projects/ORB_SLAM2/src/System.cc: In member function ‘cv::Mat ORB_SLAM2::System::TrackStereo(const cv::Mat&, const cv::Mat&, const double&)’ :
/home/sankuai/Projects/ORB_SLAM2/src/System.cc:134:28: error: ‘usleep’ was not declared in this scope
usleep(1000);
^
/home/sankuai/Projects/ORB_SLAM2/src/System.cc: In member function ‘cv::Mat ORB_SLAM2::System::TrackRGBD(const cv::Mat&, const cv::Mat&, const double&)’ :
```

```
/home/sankuai/Projects/ORB_SLAM2/src/System.cc:185:28: error: ‘usleep’ was not declared in this scope
usleep(1000);
```

参见：[ORB_SLAM2/issues/317]

[Fixed compilation error on usleep by mpdmanash · Pull Request #144 · raulmur/ORB_SLAM2 · GitHub]

可以尝试：

> Instead of adding `#include <unistd.h>` to every .cc files, you can put it in `System.h` instead. All other files include `System.h` in a nested manner.

3. 编译 ORB_SLAM2出现如下错误 `static assertion failed: std::map must have the same value_type as its allocator static_assert(is_same<typename _Alloc::value_type, value_type>::value`

```
Build type: Release
-- Using flag -std=c++11.
-- Configuring done
-- Generating done
-- Build files have been written to: /home/wang/SLAM/src/ORB_SLAM2/build
Scanning dependencies of target ORB_SLAM2
[  3%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/System.cc.o
[  6%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Tracking.cc.o
[  9%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/LocalMapping.cc.o
[ 12%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/LoopClosing.cc.o
/home/wang/SLAM/src/ORB_SLAM2/src/LoopClosing.cc: In member function ‘void ORB_SLAM2::LoopClosing::CorrectLoop()’:
/home/wang/SLAM/src/ORB_SLAM2/src/LoopClosing.cc:417:20: warning: use of an operand of type ‘bool’ in ‘operator++’ is deprecated [-Wdeprecated]
        mnFullBAIdx++;
                  ^~
In file included from /usr/include/c++/8/map:61,
                 from /home/wang/SLAM/src/ORB_SLAM2/Thirdparty/DBoW2/DBoW2/BowVector.h:14,
                 from /home/wang/SLAM/src/ORB_SLAM2/include/Frame.h:27,
                 from /home/wang/SLAM/src/ORB_SLAM2/include/MapPoint.h:25,
                 from /home/wang/SLAM/src/ORB_SLAM2/include/KeyFrame.h:24,
                 from /home/wang/SLAM/src/ORB_SLAM2/include/LoopClosing.h:24,
                 from /home/wang/SLAM/src/ORB_SLAM2/src/LoopClosing.cc:21:
/usr/include/c++/8/bits/stl_map.h: In instantiation of ‘class std::map<ORB_SLAM2::KeyFrame*, g2o::Sim3, std::less<ORB_SLAM2::KeyFrame*>, Eigen:
:aligned_allocator<std::pair<const ORB_SLAM2::KeyFrame*, g2o::Sim3> > >’:
/home/wang/SLAM/src/ORB_SLAM2/src/LoopClosing.cc:439:21:   required from here
/usr/include/c++/8/bits/stl_map.h:122:21: error: static assertion failed: std::map must have the same value_type as its allocator
       static_assert(is_same<typename _Alloc::value_type, value_type>::value,
                     ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
CMakeFiles/ORB_SLAM2.dir/build.make:134: recipe for target 'CMakeFiles/ORB_SLAM2.dir/src/LoopClosing.cc.o' failed
make[2]: *** [CMakeFiles/ORB_SLAM2.dir/src/LoopClosing.cc.o] Error 1
CMakeFiles/Makefile2:178: recipe for target 'CMakeFiles/ORB_SLAM2.dir/all' failed
make[1]: *** [CMakeFiles/ORB_SLAM2.dir/all] Error 2
Makefile:83: recipe for target 'all' failed
make: *** [all] Error 2
```

参见 [ORB-SLAM2编译错误_89: recipe for target 'frontend/cmakefiles/fronten-CSDN博客]

在虚拟机编译 ORB_SLAM 3 可能遇到的问题

编译ORB_SLAM3/ThirdParty 中自带的 Sophus 出错

```
11:14:20] ysy@ubuntu /home/ysy/Desktop/code/ORB_SLAM3_detailed_comme
 cd build/
 make -j2
  4%] Building CXX object test/core/CMakeFiles/test_so2.dir/test_so2
cpp.o
 12%] Built target test_common
 16%] Building CXX object test/core/CMakeFiles/test_se2.dir/test_se2
cpp.o
n file included from /home/ysy/Desktop/code/ORB_SLAM3_detailed_comme
ts/Thirdparty/Sophus/test/core/test_so2.cpp:3:
home/ysy/Desktop/code/ORB_SLAM3_detailed_comments/Thirdparty/Sophus/
ophus/so2.hpp:106:40: error: ‘ScalarBinaryOpTraits’ in namespace ‘Ei
en’ does not name a template type
 106 |    using ReturnScalar = typename Eigen::ScalarBinaryOpTraits<
     |                                        ^~~~~~~~~~~~~~~~~~~~~
home/ysy/Desktop/code/ORB_SLAM3_detailed_comments/Thirdparty/Sophus/
ophus/so2.hpp:110:26: error: ‘ReturnScalar’ was not declared in this
scope; did you mean ‘GetScalar’?
 110 |    using SO2Product = SO2<ReturnScalar<OtherDerived>>;
     |                           ^~~~~~~~~~~~
     |                           GetScalar
```

通过 locate 查询到电脑装了两个 Eigen 库，查看 Eigen 版本：

一个位于 `/usr/local/include/eigen3/Eigen` 为 3.2.0，另一个位于 `/usr/include/eigen3/Eigen` 为 3.3.7。它默认寻找了Eigen 3.2.0 那个，与Sophus 所需版本不匹配。

解决方式：

设置 `Eigen3_ROOT` 指向Eigen 3.3.7 对应的安装目录，使cmake 查找到合适eigen 库版本

```
# Find Eigen 3 (dependency)
set(Eigen3_ROOT "/usr/include/eigen3")
find_package(Eigen3 3.3 REQUIRED )
```

添加 `set(Eigen3_ROOT "/usr/include/eigen3")`，同时将此行添加到顶层 `CMakeLists. txt` 文件的开头：

```
cmake_policy(SET CMP0074 NEW)
```

注意：

ORB_SLAM 2 编译需要Eigen 3.2/3.1 版本，ORB_SLAM 3 编译需要Eigen 3.3 版本