

Robotics Lab 4

SA18225293 彭辰铭

实验步骤

1. ORB_SLAM2 代码下载

进入ORB_SLAM2代码托管地址

文档说明依赖库有Pangolin、Eigen3、OpenCV（2.4.3版本及以上）、DBow2、g2o。

由于之前的实验中除了DBow2，其他已经安装过了，所以这里先安装DBow2

```
1 git clone git@github.com:dorian3d/DBow2.git
2 cd DBow2
3 mkdir build
4 cd build
5 cmake ..
6 make
7 sudo make install
```

然后再下载该项目代码,并build

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

2. 解析实验代码

查看起include，src，Examples下的文件，发现其典型依赖库如上所说

有Pangolin、Eigen3、OpenCV、DBow2、g2o

3. 在KITTI数据集上运行ORB_SLAM2

首先在[官网](#)下载数据集,由于其数据集都比较大，这里我是采用了同学提供在群里的第一个数据集的01部分，来进行实验的。

进入ORB_SLAM2文件夹，运行文件

```
1 ./Examples/Monocular/mono_kitti Vocabulary/ORBvoc.txt Examples/Monocular/KITTI00-02.yaml ~/dataset/sequences/SEQUENCE_NUMBER
```

运行结果为



4.通过本地视频文件运行SLAM系统

将提供的实验代码拷贝至，ORB_SLAM2/Examples/Monocular文件夹中

修改主文件夹下的CMakeLists.txt文件,添加

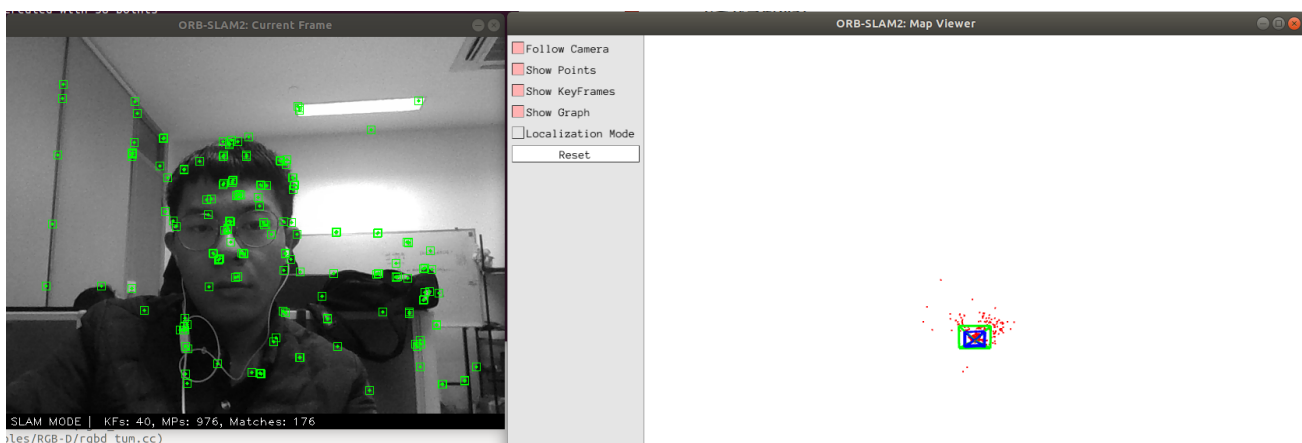
```
add_executable(mono_c920 Examples/Monocular/myslam.cpp)
target_link_libraries(mono_c920 ${PROJECT_NAME})
```

重新build

然后运行

```
1 | ./mono_c920 myslam.yaml
```

结果为



继续修改主文件夹下的CMakeLists.txt文件

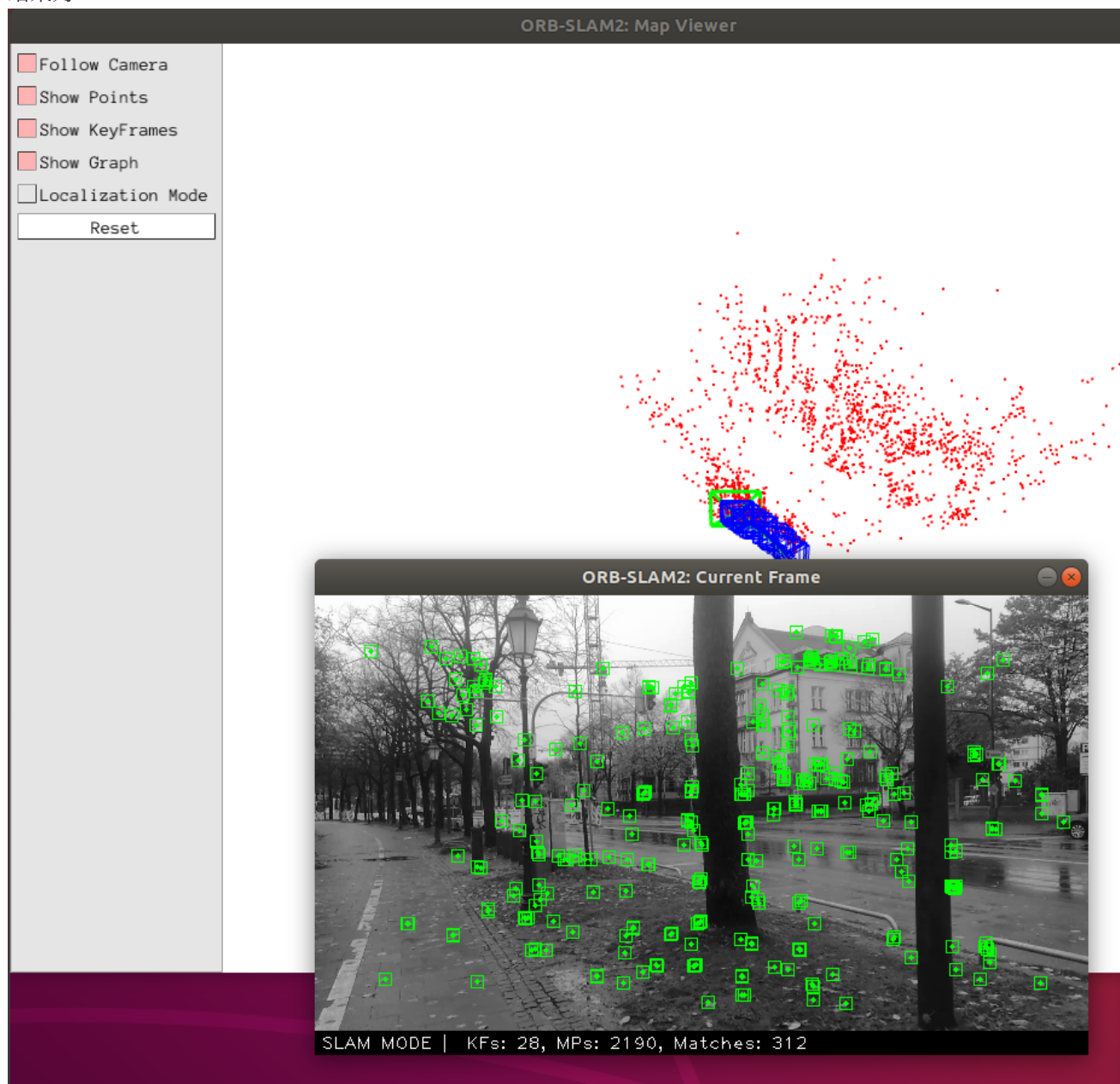
添加

```
add_executable(myvideo Examples/Monocular/myvideo.cpp)
target_link_libraries(myvideo ${PROJECT_NAME})
```

然后运行

```
1 | ./myvideo myvideo.yaml
```

结果为



整个视觉SLAM流程包括：

- 1、传感器信息读取。相机图像信息的读取和预处理
- 2、视觉里程计。视觉里程计的任务是估算相邻图像间相机的运动，以及局部地图的样子。VO又称为前端（Front End）。
- 3、后端优化。后端接受不同时刻视觉里程计测量的相机位姿，以及回环检测的信息，对它们进行优化，得到全局一致的轨迹和地图。由于接在VO之后，又称为后端（Back End）。
- 4、回环检测。回环检测判断机器人是否到达过先前的位置。如果检测到回环，它会与信息提供给后端进行处理。
- 5、建图。它根据估计的轨迹，建立与任务要求对应的地图。