

实验 4 ORB_SLAM2 实验

实验目的：

了解 SLAM 的理论和方法，视觉 SLAM 框架和主流开源系统，掌握 ORB_SLAM2 的组织架构，编译并分别在 KITTI 数据集和本地视频上运行该 SLAM 系统，获得地图输出。

实验内容：

1. ORB_SLAM2 下载与编译
2. ORB_SLAM2 解析
3. 在 KITTI 数据集上运行 SLAM
4. 在本地视频上运行 SLAM

实验设备：

自带笔记本 PC 机头，安装 Ubuntu12.04 或以上系统，OpenCV 开发库

预备知识：

1. SLAM 理论和方法
2. linux, C++编程基础, Cmake 编译器
3. OpenCV 图像处理库

实验步骤：

1. 代码下载

访问 ORB_SLAM2 代码托管地址并下载

```
stevencui@ubuntu:~/Project$ git clone https://github.com/raulmur/ORB_SLAM2.git ORB_SLAM2
Cloning into 'ORB_SLAM2'...
remote: Counting objects: 566, done.
remote: Total 566 (delta 0), reused 0 (delta 0), pack-reused 566
Receiving objects: 100% (566/566), 41.44 MiB | 10.00 KiB/s, done.
Resolving deltas: 100% (178/178), done.
Checking connectivity... done.
```

编译并执行，得到 6 个分别基于不同相机和数据集的可执行文件，1 个库文件；

2. 解析实验代码

浏览 ORB-SLAM2 中的 include, src, Examples 三个文件夹中代码，并给出该系统的典型依赖库，如 Eigen 等。

3. 在 KITTI 数据集上运行 ORB_SLAM2

按照程序说明，下载 KITTI 数据集，运行 ORB_SLAM2 并导入 KITTI 图像帧，获得相机运动估计和环境地图。

4. 通过本地视频文件运行 SLAM 系统

根据提供的样例视频和视频导入代码 myslam.cpp，修改编译描述文件 CMakeLists.txt，添加

```
add_executable(mono_c920 Examples/Monocular/myslam.cpp)

target_link_libraries(mono_c920 ${PROJECT_NAME})
```

重新编译并运行，获得在样例视频上运行后的地图输出。

实验报告

用自己的话给出上述各步骤代码框架，视觉 SLAM 理论和方法，给出实验运行结果。

文档以 Word+代码形式，放入文件夹打包提交，以“**Robotic_lab4_学号_姓名**”格式命名，每人一份，提交至教辅系统，截至时间为 2019 年 1 月 7 日。