

SLAM Theory and Practice: Home Work 1

Liang Xu
liangxucs@gatech.edu

February 2018

1 熟悉Linux

1.1 如何在Ubuntu 中安装软件(命令行界面)?它们通常被安装在什么地方?

apt-get install package name

Most of the user program saved in /usr/bin —i binary file

Sometime saved in /opt if the bin too big

1.2 linux 的环境变量是什么?我如何定义新的环境变量?

Linux 环境变量是指, Linux 运行环境的一些参数, 比如说, 临时文件的位置, bin file 的路径等。

Add the new environment variable to /etc/profile for all user

Add to ~/.bashrc for current user.

1.3 linux 根目录下面的目录结构是什么样的?至少说出3 个目录的用途

/bin :contains executable programs which are needed in single user mode

/etc :Contains configuration files which are local to the machine.

/home: home directories for users.

/lib: shared libraries

/mnt : mount point

/root:home for root user.

/usr: hold sharable, read only data.

1.4 假设我要给a.sh 加上可执行权限,该输入什么命令?

sudo chmod +x a.sh

1.5 假设我要将a.sh 文件的所有者改成xiang:xiang,该输入什么命令?

```
sudo chown xiang:xiang a.sh
```

2 SLAM Literature Survey

2.1 SLAM 会在哪些场合中用到?至少列举三个方向。

1. Can be used in the robotic vehicle to locate the current position and map the current map.
2. Use the Augmented Reality(AR) to merge the virtual and real environment.
3. 3D scene reconstruction by multiply image.

2.2 SLAM 中定位与建图是什么关系?为什么在定位的同时需要建图?

Because the sensor data always have noise part. So, no matter how accuracy the sensor is, the accumulated error will come up. SLAM solves this problem by correcting the current location and remapping the environment. The current mapping gives the prior information for the next measurement step for better position estimation. Current measurement and location can help to rebuild more accuracy map.

The map often required for other task, for example, path planning. The map also can limit the error to estimate the current sensor position.

2.3 SLAM 发展历史如何?我们可以将它划分成哪几个阶段?

SLAM has been widely used in the robotic field in the past several decade. And the history can be divided by 2 phases: classical age(1986-2004), the classical age mainly cover the probabilistic formulations. algorithm-analysis age(2004-2005), the study of fundamental properties of SLAM.

2.4 列举三篇在SLAM 领域的经典文献

1. Simultaneous Localization and Mapping: Part I , Part II
2. MonoSLAM: Real-Time Single Camera SLAM
3. FastSLAM: A factored solution to the simultaneous localization and mapping problem

3 CMake Practice

```
cmake_minimum_required( VERSION 2.8 )

project( HelloSLAM )

set( CMAKE_BUILD_TYPE "Release" )
```

```
include_directories("include")
add_library(hello SHARED src/hello.cpp)

add_executable(useHello useHello.cpp)
target_link_libraries(useHello hello)

install(TARGETS hello DESTINATION "/usr/local/lib/")
install(FILES include/hello.h DESTINATION "/usr/local/include/")
```

4 理解ORB-SLAM2 框架

4.1 Download

```
>>> git clone https://github.com/raulmur/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 | 8.80 MiB/s, done.
Resolving deltas: 100% (178/178), done.
(Liangxu:~/GT/slam/homework/opensource)-(CPU 3.4%:0)-(12K:1)
>>> ll
total 12K
4.0K drwxr-xr-x 3 liangxu liangxu 4.0K Feb 27 18:32 ./
4.0K drwxr-xr-x 4 liangxu liangxu 4.0K Feb 27 18:32 ../
4.0K drwxr-xr-x 9 liangxu liangxu 4.0K Feb 27 18:32 ORB_SLAM2/
(Liangxu:~/GT/slam/homework/opensource)-(CPU 3.4%:0)-(12K:1)
>>>
```

4.2 Answer Questions

4.2.1 ORB-SLAM2 将编译出什么结果?有几个库文件和可执行文件?

compile 19 library files in src/ and link those library with opencv, eigen3, pangolin, DBoW2 and g2o. Compilation will generate six executable files and saved in the /Example/** folder.

4.2.2 ORB-SLAM2 中的include, src, Examples 三个文件夹中都含有什么内容?

include have all the header file, and src have all the source code. Example folder contains different examples for using this research project.

4.2.3 ORB-SLAM2 中的可执行文件链接到了哪些库?它们的名字是什么?

OpenCV, computer vision
eige3, for matrix computation
Pangolin, for display

```

canning dependencies of target stereo_kitti
69%] Building CXX object CMakeFiles/stereo_kitti.dir/Examples/Stereo/stereo_kitti.cpp.o
72%] Linking CXX executable ../Examples/Stereo/stereo_kitti
usr/bin/ld: warning: libopencv_core.so.2.4, needed by ../Thirdparty/DBow2/libopencv.so.3.3, was not found in the default library search path
72%] Built target stereo_kitti
canning dependencies of target cam
75%] Building CXX object CMakeFiles/cam.dir/Examples/code/myslam.cpp.o
77%] Linking CXX executable ../Examples/code/cam
usr/bin/ld: warning: libopencv_core.so.2.4, needed by ../Thirdparty/DBow2/libopencv.so.3.3, was not found in the default library search path
77%] Built target cam
canning dependencies of target stereo_euroc
80%] Building CXX object CMakeFiles/stereo_euroc.dir/Examples/Stereo/stereo_euroc.cpp.o
83%] Linking CXX executable ../Examples/Stereo/stereo_euroc
usr/bin/ld: warning: libopencv_core.so.2.4, needed by ../Thirdparty/DBow2/libopencv.so.3.3, was not found in the default library search path
83%] Built target stereo_euroc
canning dependencies of target mono_tum
86%] Building CXX object CMakeFiles/mono_tum.dir/Examples/Monocular/mono_tum.cpp.o
88%] Linking CXX executable ../Examples/Monocular/mono_tum
usr/bin/ld: warning: libopencv_core.so.2.4, needed by ../Thirdparty/DBow2/libopencv.so.3.3, was not found in the default library search path
88%] Built target mono_tum
canning dependencies of target mono_kitti
91%] Building CXX object CMakeFiles/mono_kitti.dir/Examples/Monocular/mono_kitti.cpp.o
94%] Linking CXX executable ../Examples/Monocular/mono_kitti
usr/bin/ld: warning: libopencv_core.so.2.4, needed by ../Thirdparty/DBow2/libopencv.so.3.3, was not found in the default library search path
94%] Built target mono_kitti
canning dependencies of target mono_euroc
97%] Building CXX object CMakeFiles/mono_euroc.dir/Examples/Monocular/mono_euroc.cpp.o
100%] Linking CXX executable ../Examples/Monocular/mono_euroc
usr/bin/ld: warning: libopencv_core.so.2.4, needed by ../Thirdparty/DBow2/libopencv.so.3.3, was not found in the default library search path
100%] Built target mono_euroc

```

5 使用摄像头或视频运行ORB-SLAM2

5.1 compilation finish

5.2 CMakeLists.txt 修改方案

move the code folder under Example Then add the following code to Cmake-List.txt. Two executable files will be created, one is for recorded video called video, another is for the video camera, called cam.

```

set(CMAKE_RUNTIME_OUTPUT_DIRECTORY ${PROJECT_SOURCE_DIR}/Examples/code)
add_executable(video
Examples/code/myvideo.cpp)
target_link_libraries(video ${PROJECT_NAME})

add_executable(cam
Examples/code/myslam.cpp)
target_link_libraries(cam ${PROJECT_NAME})

```

5.3 Result

CMake give us a very robust compilation environment setup.

