

ROS OpenCV 实验讲义

【讲义中仅包含实验部分，代码的讲解会在课堂上进行】

0. 课前准备

0.1 安装 PCL 库

*****请提前自行安装 PCL 库，并安装 ros-kinetic-pcl-ros 和 ros-kinetic-pcl-msgs。建议使用源码安装 PCL。下述方法仅供参考，请自行摸索。*****

参考：<https://blog.csdn.net/dantengc/article/details/78446600> 中的方式三。个别依赖可能存在版本较老的问题而无法安装，一般情况下 tab 出新版本就可以了。

PCL 安装完成之后要安装 ros-kinetic-pcl-ros 和 ros-kinetic-pcl-msgs：

```
sudo apt-get install ros-kinetic-pcl-ros
sudo apt-get install ros-kinetic-pcl-msgs
```

0.2 安装 kinect v1 驱动

在终端运行：

```
sudo apt-get install ros-kinetic-openni-* ros-kinetic-freenect-*
```

即可，运行时连接 kinect，使用：

```
roslaunch freenect_launch freenect.launch
```

运行。

1. 摄像头驱动安装与测试

```
sudo apt-get install ros-kinetic-usb-cam
```

```
roscore
```

```
roslaunch usb_cam usb_cam_node
```

此时应该能看到摄像头指示灯亮。

另开一个终端，运行 rviz：

```
roslaunch rviz rviz
```

点击左下角“add”，选择“image”，并在左侧栏中 image 下的 Image Topic 选择“/usb_cam/image_raw”，即可显示摄像头实时画面。

同理也可以使用 rqt_image_view：

```
roslaunch rqt_image_view rqt_image_view
```

在订阅话题中选择“/usb_cam/image_raw”即可。

2. ROS 中使用 OpenCV

将 ros_vision_program 中的 opencv_use 文件夹整个拷贝到 catkin 的 src 文件夹中，并使用 catkin_make 编译：

```
cd catkin
```

```
catkin_make
```

```
source devel/setup.bash
```

假设在文件夹/home/***/image 下有一张图片 1.jpg，那么就运行：

```
roslaunch opencv_use opencv_use_node /home/***/image/1.jpg
```

然后就会显示原图像和进行灰度变换后的图像。

3. cv_bridge 的使用

结合发布和订阅器使用 cv_bridge，实现一个节点读取摄像头图像并且发布，在另外一个节点订阅，然后进行显示。发布和订阅器参考 ros wiki，cv_bridge 的使用参考 cv_bridge 的 Tutorials，ros document: http://docs.ros.org/api/cv_bridge/html/

3.1 接收节点：订阅消息

```
cd ~/catkin/src
catkin_create_pkg image_transportor cv_bridge roscpp
将 image_transportor_node.cpp 拷贝到 ~/catkin/src/image_transportor/src 中，并修改
CMakeLists.txt 为：
#从这里开始#
cmake_minimum_required(VERSION 2.8.3)
project(image_transportor)
find_package(catkin REQUIRED COMPONENTS
    cv_bridge
    roscpp
)
find_package(OpenCV REQUIRED)
catkin_package()
include_directories(
    ${catkin_INCLUDE_DIRS}
    ${OpenCV_INCLUDE_DIRS}
)
add_executable(${PROJECT_NAME}_node src/image_transportor_node.cpp)
target_link_libraries(${PROJECT_NAME}_node
    ${catkin_LIBRARIES}
    ${OpenCV_LIBRARIES}
)
#CMakeLists.txt 至此结束。注意，不要修改该源文件，尤其是注释部分，在接下来的
实验中会用到#
cd ~/catkin
catkin_make
source devel/setup.bash
roscore
roslaunch image_transportor image_transportor_node
roslaunch usb_cam usb_cam_node
此时应该可以看到一个黑白的实时画面。
```

3.2 发布与订阅

修改 image_transportor_node.cpp，去掉第 5~31 行和第 44 行的注释，将第 33~39 行、第 46、47 行注释。重新用 catkin_make 编译并 source。
roscore

```
roslaunch image_transportor image_transportor_node
roslaunch usb_cam usb_cam_node
roslaunch rqt_image_view rqt_image_view
```

此时可以在 rqt_image_view 中看到消息/image_transportor/image_edge，这就是程序在接收摄像机图像后进行拉普拉斯变换后再将该消息发布的结果。

4. ROS 中使用 PCL

请提前自行安装 PCL 库，并安装 ros-kinetic-pcl-ros 和 ros-kinetic-pcl-msgs。建议使用源码安装 PCL。

```
sudo apt-get install ros-kinetic-pcl-ros
sudo apt-get install ros-kinetic-pcl-msgs
```

提示缺什么依赖就装什么。

4.1 仅订阅深度图

创建 ros package:

```
catkin_create_pkg my_pcl_tutorial pcl_ros roscpp cv_bridge sensor_msgs
```

CMakeLists.txt:

```
cmake_minimum_required(VERSION 2.8.3)
project(my_pcl_tutorial)
find_package(catkin REQUIRED COMPONENTS
  cv_bridge
  roscpp
  sensor_msgs
  pcl_ros
)
```

```
catkin_package()
```

```
find_package(PCL REQUIRED)
find_package(OpenCV REQUIRED)
```

```
include_directories(
  ${catkin_INCLUDE_DIRS}
  ${PCL_INCLUDE_DIRS}
  ${OpenCV_INCLUDE_DIRS}
)
add_executable(${PROJECT_NAME}_node src/my_pcl_tutorial_node.cpp)

target_link_libraries(${PROJECT_NAME}_node
  ${catkin_LIBRARIES}
  ${OpenCV_LIBRARIES}
  ${PCL_LIBRARIES}
)
```

将 my_pcl_tutorial_node.cpp 拷贝到/catkin/src/my_pcl_tutorial/src 中，之后 catkin_make 编译。

假设 RGB-D 数据集 rgb_d_dataset_freiburg1_desk.bag 【下载自 TUM Dataset 官网，可以自行下载，也可以现场拷贝】位于/home/xxx/download 中：

```
roscore
roslaunch my_pcl_tutorial my_pcl_tutorial_node
rviz
rosbag play /home/xxx/download/rgb_d_dataset_freiburg1_desk.bag
```

并在 rviz 中添加 pointcloud2，topic 选择/output，此时应该可以观察到一个黑色的点云图。

4.2 显示彩色点云图：消息同步

将全部有效代码注释掉，去掉下面注释了的代码的注释（第 77 行及以下内容），重新 catkin_make 后运行，这时应该可以看到彩色的点云图。

5. ROS 中使用 kinect 相机复原点云图

将 turtlebot 的 kinect 相机连接至笔记本电脑的 usb 口，打开 turtlebot 电源，运行：

```
roslaunch freenect_launch freenect.launch
```

启动相机驱动，并运行实验 4 中的 my_pcl_tutorial_node 节点：

```
roslaunch my_pcl_tutorial my_pcl_tutorial_node
```

打开 rviz，在左侧栏目中添加 pointcloud2，topic 仍然选择/output，此时应该可以观察到实时的彩色点云图。使用手柄或手动移动 turtlebot，可以看到点云图的实时变化。