

Ubuntu18.04 环境配置说明文档2023

Ubuntu全环境配置说明文档

编写人：卢佳兴

编制时间：2021年11月15日

修改时间：2022/5、27

2023.4.20

V1.0环境控制文档上增加了传感器以及图像识别算法的配置方法，从基于Ubuntu16.04的控制系统升级到了Ubuntu18.04，已经经过测试在实验校车上运行。

具备ubuntu18.04系统。

部分情况下github下载（clone）代码的速度较慢，可以采取gitee方式进行下载

Github：

熟悉一些shell基础语言，cd、mkdir、rm...

0 前期准备：

0.1 ros的安装过程

ros需要安装melodic版本

点击连接访问：[melodic/Installation - ROS Wiki](#)

0.2 显卡驱动的安装

参考[Ubuntu 18.04 安装 NVIDIA 显卡驱动 - 知乎 \(zhihu.com\)](#)

具体步骤：

ubuntu-drivers devices

sudo ubuntu-drivers autoinstall 或者使用 sudo apt install nvidia-465

1 环境配置

1.1 catkin build工具的配置

如果在配置ros时不是用实验室的脚本配置需要在配置完ros之后在终端输入下面的命令，用于catkin build等工具套件的配置。实验室规划控制使用catkin build进行二进制文件的编译，而非catkin_make。

```
sudo apt-get install python-catkin-tools
```

1.2 安装各个库

在主目录下新建文件夹专门存放各个库的源码文件，进入该文件夹

(1) 安装Ceres库

github上下载库源码文件夹

```
git clone -b 1.12.0 https://gitee.com/mirrors/ceres-solver.git
```

```
cd ceres-solver
```

首先安装依赖：

```
sudo apt-get install cmake libgoogle-glog-dev libatlas-base-dev libsuitesparse-dev
```

安装ceres库：

```
mkdir build&&cd build
```

```
cmake ..
```

make -j8 (执行此命令会等待很久，编译过程十分缓慢) (cpu四核八线程为-j8,六核十二线程-j12,八核十六线程-j16，编译速度最快)

```
sudo make install
```

安装完成

(2) 安装nlopt非线性优化库

首先回到装源各个文件的文件夹

下载库源码文件夹：

```
git clone https://gitee.com/cuizhi/nlopt.git
```

安装依赖：

```
sudo apt-get -y install cmake g++
```

进入nlopt源码文件夹：

```
cd nlopt
```

正式安装库：

```
mkdir build&&cd build
```

```
cmake ..
```

```
make -j8
```

```
sudo make install
```

(3) 安装cpp_ad（官方可参考<https://coin-or.github.io/CppAD/doc/install.htm>）

下载库源码文件：

```
git clone https://github.com/coin-or/CppAD.git
```

```
cd cppad(根据文件夹名进入文件夹)
```

正式安装：

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make
```

```
sudo make install
```

(4) 安装flexible-collision-library库

```
sudo chmod u+x install_libccd_fcl.sh
```

```
./install_libccd_fcl.sh
```

(5) 安装pcl以及栅格地图相关库

```
sudo apt-get update
```

```
sudo apt-get -y install ros-melodic-pcl-ros ros-melodic-costmap-2d ros-melodic-grid-map
```

(6) 安装配合vrep使用的一个用于坐标的库

```
sudo apt-get update
```

```
sudo apt-get -y install libgeographic-dev
```

(7) 安装Google跨平台检测的库

```
sudo apt-get update
```

```
sudo apt-get install -qq libgtest-dev
```

h. snopt安装

<https://ccom.ucsd.edu/~optimizers/downloads/>

采用脚本安装snopt需要在bit_ivrc组里。

主要是由于脚本这一句需要将snopt_binary下载下来。

```
git clone git@github.com:bit-ivrc/snopt\_binary.git && cd snopt_binary.
```

目前配置文件在“环境配置-2.4snopt-snopt_binary-master” 将配置文件解压至系统\$HOME下然后解压。

```
cd snopt_binary-master;
```

```
bash install.sh
```

1.3 通过命令行安装的库文件

(1) 安装map_server用到的消息类型库：

```
sudo apt-get install ros-melodic-tf2-sensor-msgs
```

(2) 安装速度规划相关

```
sudo apt-get install ros-melodic-roscpp
```

(3) 安装局部规划相关

```
sudo apt-get install ros-melodic-grid-map-sdf
```

```
sudo apt-get install ros-melodic-tf2-sensor-msgs ros-melodic-roscpp ros-melodic-grid-map-sdf
```

1.4 其他已经下载的库文件的安装方法

基本的安装过程如下所示，进入到软件安装包的主目录下

```
mkdir build && cd build
```

```
cmake ..
```

```
make -j8
```

```
sudo make install
```

安装的主要文件有：

- (1) controllib
- (2) Bench_mark
- (3) Casadi

这些文件已经有安装包，可以在Ubuntu18.04中使用

2 ipopt的安装

首先安装依赖：

```
sudo apt-get install libblas3 libblas-dev liblapack3 liblapack-dev gfortran build-essential  
cppad gcc g++
```

ipopt安装直接参考：https://blog.csdn.net/qq_24649627/article/details/103084849

ubuntu18.04 需要安装 ipopt 3.12.7/4版本

部分库下载不下来，参考<https://github.com/coin-or-tools>位置

3 snopt安装

主要是由于脚本这一句需要将snopt_binary下载下来。

```
git clone git@github.com:bit-ivrc/snopt\_binary.git && cd snopt_binary。
```

目前配置文件在“环境配置-2.4snopt-snopt_binary-master” 将配置文件解压至系统\$HOME下然后解压。

```
git clone git@github.com:bit-ivrc/snopt\_binary.git && cd snopt_binary。
```

下载不下来可以直接用压缩包提取

```
cd snopt_binary-master;
```

```
bash install.sh
```

4 vrep安装

“环境配置- 2.6vrep安装- V-REP_PRO_EDU_V3_5_0_Linux.tar.gz”

将配置文件先放置到\$HOMEapplication下：

```
tar -zxvf V-REP_PRO_EDU_V3_5_0_Linux.tar.gz
```

```
sudo chmod u+x vrep.sh
```

```
echo "alias vrep = \"$HOME /application/V-REP_PRO_EDU_V3_5_0_Linux/vrep.sh\"" >>
~/.bashrc
```

```
source ~/.bashrc
```

终端输入vrep，如果vrep启动就证明vrep安装成功

5 传感器安装

5.1 红外相机

红外相机为usb驱动及供电，即插上usb接口之后相机便可以成像并传递数据，驱动的核心usb_camera节点的调用。

首先得到usb_cam的源代码文件，放入新建的catkin工作空间src文件夹下，返回主文件夹，运行命令catkin_make，通过编译，创建usb_cam的节点。

也可以直接从github上下载usb_cam源码，从创建工作空间到下载源码并编译全过程代码如下：

```
mkdir -p vision_WS/src
```

```
cd vision_WS
```

```
catkin_make
```

```
cd src
```

```
git clone https://github.com/ros-drivers/usb\_cam.git
```

```
cd ..
```

```
catkin_make
```

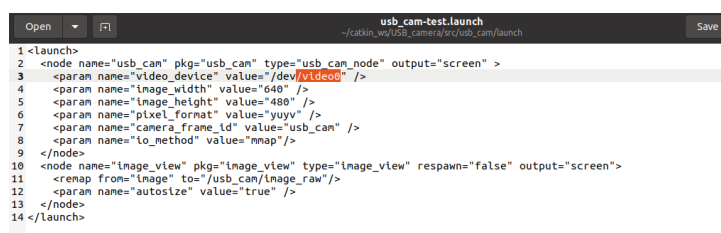
如上即可得到编译通过的usb_cam节点，接下来调用节点

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

```
roslaunch usb_cam usb_cam-test.launch
```

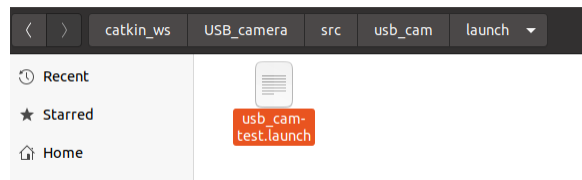
即可启动节点。

默认情况下，第一次启动节点将调用电脑摄像头，调用哪个相机可以在USB_camera/src/usb_cam/launch/usb_cam-test.launch文件里更改，尝试将video0更改为video1，video2，video3...即可以调出红外相机的实时图像。



```
1 <launch>
2   <node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen" >
3     <param name="video_device" value="/dev/video0" />
4     <param name="image_width" value="640" />
5     <param name="image_height" value="480" />
6     <param name="pixel_format" value="yuyv" />
7     <param name="camera_frame_id" value="usb_cam" />
8     <param name="to_method" value="mmap" />
9   </node>
10  <node name="image_view" pkg="image_view" type="image_view" respawn="false" output="screen">
11    <renap from="image" to="/usb_cam/image_raw"/>
12    <param name="autosize" value="true" />
13  </node>
14 </launch>
```

如有疑问http://wiki.ros.org/usb_cam上有对该节点的详细说明。

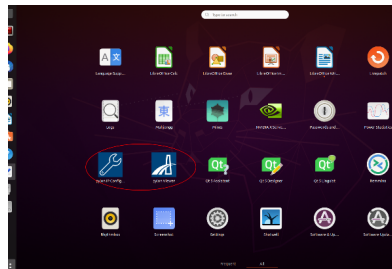


5.2 工业basler相机驱动

首先要安装pylon官方SDK，有两个软件，一个命令行即可。

```
sudo dpkg -i pylon_6.1.1.19861-deb0_amd64.deb
```

安装好之后如右图所示。



该相机驱动的源码文件夹来自<https://github.com/basler/pylon-ros-camera.git>（具体过程参考<https://blog.csdn.net/CCCranner/article/details/107856943?spm=1001.2014.3001.5501>）

进入工作空间vision_WS 文件夹

```
cd src
```

```
git clone https://github.com/basler/pylon-ros-camera.git
```

```
git clone https://github.com/dragandbot/dragandbot_common.git
```

```
sudo sh -c 'echo "yaml https://raw.githubusercontent.com/basler/pylon-ros-camera/master/pylon_camera/rostopic/pylon_sdk.yaml" > /etc/ros/rosdep/sources.list.d/30-pylon_camera.list'
```

```
rosdep update
```

```
sudo rosdep install --from-paths . --ignore-src --rosdistro=$ROS_DISTRO -y
```

```
cd ..
```

```
catkin_make
```

至此编译完成，可通过以下命令调用节点

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

```
roslaunch pylon_camera pylon_camera_node.launch
```

5.3 激光雷达驱动

激光雷达驱动有多个版本，新版本的驱动代码不适用于旧版本的雷达，因此应先结合雷达的生产日期去选取同期推出的驱动代码。本雷达生产于2019年，暂且选用rs_lidar1.1.2版本，目前尚有一些问题没有解决，但是驱动方法一致。

下载 rs_lidar驱动

git clone https://github.com/robosense-lidar/rslidar_sdk

将源码文件放入工作空间src当中

先配置必须环境：

sudo apt-get update

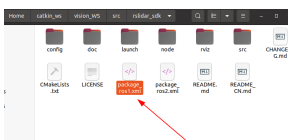
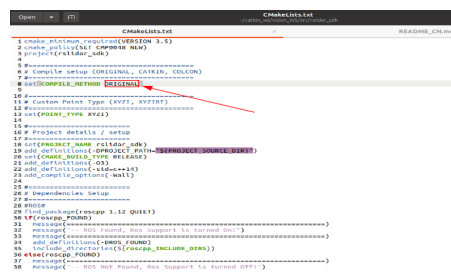
sudo apt-get install -y libyaml-cpp-dev

sudo apt-get install -y libpcap-dev

sudo apt-get install -y libprotobuf-dev protobuf-compiler

然后在源码文件夹进行操作：

打开工程内的*CMakeLists.txt*文件，将文件顶部的**set(COMPILER_METHOD ORIGINAL)**改为**set(COMPILER_METHOD CATKIN)**。



将rslidar_sdk工程目录下的
*package_ros1.xml*文件重命名为
package.xml

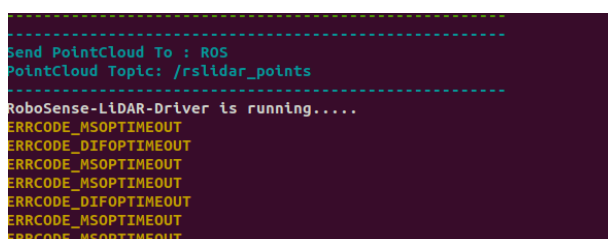
回到工作空间目录，运行以下代码即可在rviz中显示3D点云

catkin_make

source devel/setup.bash

roslaunch rslidar_sdk start.launch

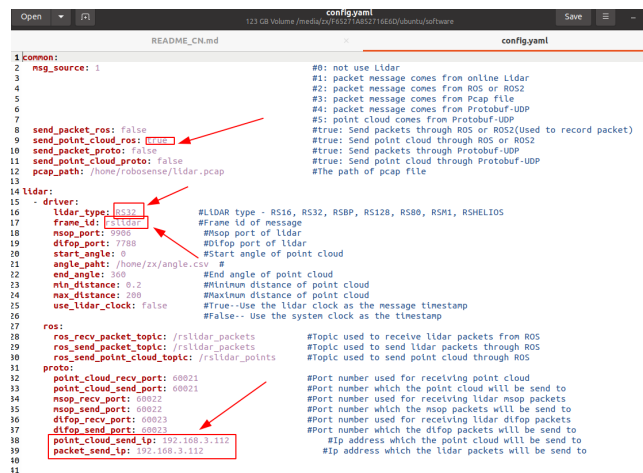
如果出现以下报错：



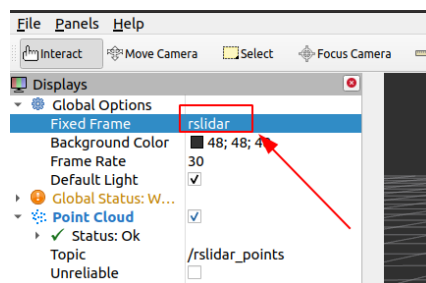
是ip端口的配置出现了问题，针对该32线激光雷达，应在config.yaml文件中进行更改。



改后如下：



端口号 msop_port 需要使用wireshark软件抓取数据包确定端口号。Point_cloud_send_ip是激光雷达出厂后直接设置的ip地址。



最后在rviz中更改Fixed Frame参数即可显示点云。

6 激光里程计与建图

由于原先slam只支持ubuntu16.04，使用aloam激光雷达slam进行替换

具体原理可见麻省理工大学张继博士的论文LOAM: Lidar Odometry and Mapping in Real-time。

具体操作如下：

在工作空间src文件夹下

git clone https://github.com/laboshinl/loam_velodyne.git

cd ..

catkin_make

即可编译源码，然后运行以下任意一条命令即可调用源码，两个launch为不同的原理，详见说明文档。

roslaunch loam_velodyne hector_loam_velodyne.launch

roslaunch loam_velodyne loam_velodyne.launch

然后对rosbag进行处理，在rosbag文件夹下

roscore

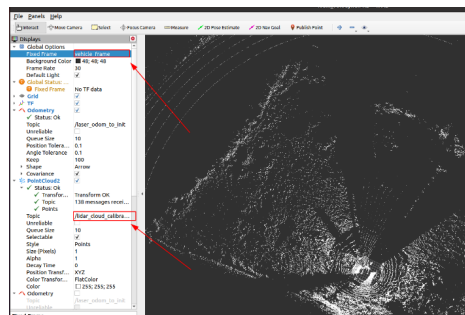
rosbag info yourname.bag

```
root@ubuntu:~/catkin_ws/src/loam_velodyne/launch$ rosbag info lidar_2021-08-25-09-35-53_0.bag
path:          lidar_2021-08-25-09-35-53_0.bag
version:       2.0
duration:      2:45s (165s)
start:         Aug 25 2021 09:35:53.53 (1629855353.53)
end:           Aug 25 2021 09:38:38.83 (1629855518.83)
size:          14.4 GB
messages:      1645
compression:   none [1640/1640 chunks]
types:         sensor_msgs/PointCloud2 [1158d486dd51d683ce2f1be655c3c181]
topics:        /lidar_cloud_calibrated 1645 msgs : sensor_msgs/PointCloud2
```

显示如下:

以上是rosbag的信息，输入以下代码即可跑数据集：

rosbag play yourname.bag



最后在rviz中修改参数即可显示激光雷达点云并建图。

7 图像感知的深度学习安装

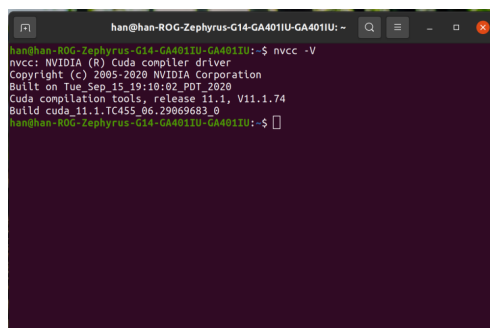
在Ubuntu18.04下进行目标检测基于Yolo v5 算法，目前已经安装了显卡驱动，Yolo v5的需要安装pytorch依赖。

安装cuda和cudnn直接按照nvidia官网上进行安装，推荐安装cuda10.1，如果显卡是RTX 30系列显卡需要安装cuda11。

输入命令：

Nvcc -V

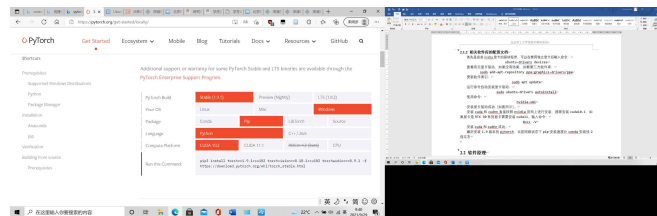
测试得到（如图 4所示） 安装cuda和cudnn成功。



需要安装anaconda，并新建anaconda环境安装pytorch

conda create -n mypytorch python=3.8

最后安装1.9版本的pytorch（如图 5所示），当前网络状态下pip安装速度比conda安装快2倍左右，官网地址是：
<https://pytorch.org/get-started/locally/>。



完成安装后在终端输入如下命令：

```
Import torch
```

```
torch.cuda.is_available()
```

测试得到（如图 6所示） pytorch1.9安装成功。

```
han@han-ROG-Zephyrus-G14-GA401IU-GA401IU: ~  
>>> (pytorch1.9) han@han-ROG-Zephyrus-G14-GA401IU-GA401IU:~$ python  
Python 3.8.11 (default, Aug 3 2021, 15:09:35)  
[GCC 7.5.0] :: Anaconda, Inc. on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import torch  
>>> torch.cuda.is_available()  
File "<stdin>", line 1  
    torch.cuda.is_available()  
      ^  
SyntaxError: invalid syntax  
>>> KeyboardInterrupt  
>>> (pytorch1.9) han@han-ROG-Zephyrus-G14-GA401IU-GA401IU:~$ python  
Python 3.8.11 (default, Aug 3 2021, 15:09:35)  
[GCC 7.5.0] :: Anaconda, Inc. on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import torch  
>>> torch.cuda.is_available()  
<function is_available at 0x7f81442ef9d0>  
>>> torch.cuda.is_available()  
True  
>>>
```

```
cd catkin_ws/src
```

```
git clone https://github.com/qq44642754a/Yolov5\_ros.git
```

```
cd yolov5_ros/yolov5
```

```
sudo pip install -r requirements.txt
```

```
roslaunch yolov5_ros yolo_v5.launch
```