1. Install ubuntu 16.0.4

1.1 Install ubuntu16.04

In windows, 准备适当的没有分区的空白空间（通过压缩window的分区），安装ubuntu时会自动识别并安装在该空白空间上

Plugin the installation drive prepared by our lab

Reboot system and keep pressing F12 to enter the Bios interface

Choose boot from the ubuntu installation drive and install as instructed

1.2 Install Sogou

sudo apt-get install fcitx libssh2-1   
从搜狗官方下载deb包   
sudo dpkg -i sogoupinyin\_2.2.0.0108\_amd64.deb

sudo apt-get install -f

sudo dpkg -i sogoupinyin\_2.2.0.0108\_amd64.deb

系统设置>语言支持>键盘输入方式系统   
然后选择 fcitx 项   
重启系统

Goto System Settings → Text Entry

点击左下角的+号，在打开的窗口中找到搜狗输入法Sogou pinyin点击Add添加进去

Goto Input Method Configuration,

点击左下角的+号，在打开的窗口中找到搜狗输入法Sogou

2. Install runfile Nvidia driver (skip 2&3&4 if without GPU)

nouveau禁止命令写入文件

sudo vim /etc/modprobe.d/blacklist.conf   
blacklist nouveau   
blacklist lbm-nouveau   
options nouveau modeset=0   
alias nouveau off   
alias lbm-nouveau off

调用指令禁止nouveau

echo options nouveau modeset=0 | sudo tee -a /etc/modprobe.d/nouveau-kms.conf

建立新的内核

sudo update-initramfs -u

sudo reboot

进入tty模式 ctrl + alt+ F1

sudo service lightdm stop   
sudo init 3

cd nvidiadirectory/   
chmod +x NVIDIA-Linux-x86\_64-390.67.run   
sudo sh NVIDIA-Linux-x86\_64-390.67.run --no-opengl-files

sudo service lightdm start

3. Install runfile cuda toolkit

因为驱动之前已经安装，这里就不要选择安装驱动。其余的都直接默认或者选择是即可

sudo sh cuda\_9.0.176\_384.81\_linux.run --override --silent --toolkit # 安装的cuda在/usr/local/cuda下面

方法二：

[https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#runfile-uninstallation](https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html" \l "runfile-uninstallation)

首先卸载之前的run file安装的版本：sudo /usr/local/cuda-10.1/bin/cuda-uninstaller

4. Install cuDNN

cd /usr/local/cuda # cuDNN放在这个目录下解压

tar -xzvf cudnn-9.0-linux-x64-v7.0.tgz

sudo cp cuda/include/cudnn.h /usr/local/cuda/include

sudo cp cuda/lib64/libcudnn\* /usr/local/cuda/lib64

sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn\*

export LD\_LIBRARY\_PATH="$LD\_LIBRARY\_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64"

export CUDA\_HOME=/usr/local/cuda

即将上述代码放入~/.bashrc文件保存后source ~/.bashrc

测试cuda的Samples

cd /usr/local/cuda-9.0/samples/1\_Utilities/deviceQuery

sudo make

./deviceQuery123

如果显示的是一些关于GPU的信息，则说明安装成功了

检查安装版本

cuda版本：

cat /usr/local/cuda/version.txt

cudnn版本

cat /usr/local/cuda/include/cudnn.h | grep CUDNN\_MAJOR -A 2

5. Install relevant python packages

sudo apt-get remove python-pip

sudo apt-get install curl

curl <https://bootstrap.pypa.io/get-pip.py> -o get-pip.py

sudo python get-pip.py

sudo pip install pip==9.0.1

sudo pip install numpy==1.14.5 scipy scikit-learn scikit-image opencv-python keras ipython==5.5.0 pyserial==3.4 enum34==1.1.6 futures==3.1.1 html5lib==0.9999999 cython==0.25.2

sudo apt-get install python-catkin-tools python-cv-bridge

6. Install tensorflow (skip if don’t use deep learning)

#sudo pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/ https://mirrors.tuna.tsinghua.edu.cn/tensorflow/linux/gpu/tensorflow\_gpu-1.4.0-cp27-none-linux\_x86\_64.whl

sudo pip3 install tensorflow-gpu==1.8.0

7. Install pcl and python-pcl (skip if don’t use point cloud library)

Install pcl for 1.8.1 from source, apt-get will install ros pcl for 1.7.1

PCL, where the ConditionalRemoval constructor has been removed (where it was previously just marked deprecated). Fixed by cloning PCL from the 1.8.1 tag.

cd ~

wget https://github.com/PointCloudLibrary/pcl/archive/pcl-1.8.1.tar.gz

tar -xf pcl-1.8.1.tar.gz

cd pcl-pcl-1.8.1 && mkdir build && cd build

cmake ..

make

sudo make install

sudo reboot

sudo pip install cython==0.25.2

Remove the following apt installed package and reinstalled them from source code in catkin work space:

sudo apt-get remove ros-kinetic-pcl-msgs ros-kinetic-pcl-conversions ros-kinetic-pcl-ros

cd {PATH\_TO\_YOUR\_CATKIN\_WORKSPACE}/src

git clone https://github.com/ros-perception/perception\_pcl.git

git clone https://github.com/ros-perception/pcl\_msgs.git

catkin clean

catkin build

Install python-pcl:

git clone https://github.com/as-wanfang/python-pcl.git

Issue: Package pcl\_2d-1.8 was not found in the pkg-config search path.

Solution: https://github.com/strawlab/python-pcl/issues/97

Remove pcl\_2d-1.8 in /usr/local/lib/pkgconfig/pcl\_features-1.8.pc ( remove ./build and rebuild)

cd python-pcl

python setup.py build

[sudo python setup.py install](https://github.com/strawlab/python-pcl.git)

Issue: ImportError: libpcl\_segmentation.so.1.8: cannot open shared object file: No such file or directory

Solution: vim ~/.bashrc and add "export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64:/usr/local/cuda-9.0/lib64:/usr/local/lib"

8. Install ROS (make sure have good internet connection)

Go setting→Ubuntu Software → Download from

Change to source website to <http://mirrors.tuna.tsinghua.edu.cn/ubuntu>

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116

sudo apt-get update

sudo apt-get install ros-kinetic-desktop-full

sudo rosdep init

rosdep update

echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc

source ~/.bashrc

sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential

sudo apt-get install ros-kinetic-moveit

source /opt/ros/kinetic/setup.bash

9. Install dexnet 3.0

sudo pip install trimesh meshrender

# The newest version 1.15.0 has problems

sudo pip install numpy==1.14.5

#cd /catkin\_ws/src/autolab\_core

#sudo pip install -e .

#cd /catkin\_ws/src/perception

#sudo pip install -e .

#cd /catkin\_ws/src/visualization

#sudo pip install -e .

cd /catkin\_ws/

git clone https://github.com/BerkeleyAutomation/autolab\_core.git

git checkout learning-tools

git clone https://github.com/BerkeleyAutomation/perception.git

git clone <https://github.com/BerkeleyAutomation/visualization.git>

git clone https://github.com/as-wanfang/gqcnn.git

git checkout merge-jeff

rosdep install --from-paths src --ignore-src --rosdistro=kinetic -y

catkin clean

catkin build

add the following to ~/.bashrc:

export GAZEBO\_MODEL\_PATH=~/catkin\_ws/src/RoboND-Perception-Project/pr2\_robot/models:$GAZEBO\_MODEL\_PATH

source ~/catkin\_ws/devel/setup.bash

source ~/.bashrc

10. Install universal\_robot and ur modern driver

This fork has used ur\_modern\_driver and adds kinect camera in gazebo

cd /catkin\_ws/src/

git clone <https://github.com/as-wanfang/universal_robot.git>

git checkout kinetic

cd ~/catkin\_ws/

catkin build

source devel/setup.bash

Step 1: launch robot

Actual robot:

roslaunch ur\_modern\_driver ur5\_bringup.launch limited:=true robot\_ip:=192.168.1.103

roslaunch ur\_description pick\_place\_project.launch limited:=true

Simulation in Gazebo

roslaunch ur\_gazebo ur5.launch limited:=true

Step 1: Control with MoveIt

roslaunch ur5\_moveit\_config ur5\_moveit\_planning\_execution.launch limited:=true (sim:=true)

roslaunch ur5\_moveit\_config moveit\_rviz.launch config:=true

the actual transform between base and tool0\_controller will not be published by the robot\_state\_publisher but will be taken from this driver via /tf.

rostopic echo /tf

Issues:

<https://github.com/ros/robot_model/issues/188>

!!Converting unknown joint type of joint 'base\_tool0\_controller\_floating\_joint' into a fixed joint

kdl\_parser is not [kdl\_parser.cpp:89](https://github.com/ros/robot_model/blob/409c4b923c38582ba595c7fc2a0536f0ee60045c/kdl_parser/src/kdl_parser.cpp" \l "L89). Planar joints are also missing from kdl\_parser. I think the root issue is KDL only supports joints with 1 degree of freedom. [Documentation here](http://docs.ros.org/api/orocos_kdl/html/classKDL_1_1Joint.html" \l "ad05f78142c8b73c3977f038a7a2c544e)

!!Unable to update multi-DOF joint 'base\_tool0\_controller\_floating\_joint': TF has no common time between 'base' and 'tool0\_controller'

The second error in your output is from [robot\_state\_publisher.cpp:66](https://github.com/ros/robot_state_publisher/blob/b9da64fd509521c9f24f9be634054d8366174180/src/robot_state_publisher.cpp" \l "L66). It was introduced in [this commit](https://github.com/ros/robot_state_publisher/commit/8235bce9b65ee9723a4f0a0e03d7258737cf07ff). The error message seems to say won't publish the transform because [sensor\_messages/JointState](http://docs.ros.org/api/sensor_msgs/html/msg/JointState.html) doesn't have enough information to publish the transform. robot\_state\_publisher expects to publish updated joint transforms based on input on it's topic joint\_states

joint\_state\_publisher is to publish sensor\_msgs/JointState messages, but that message type cannot describe floating joint (more than 1 degree of freedom)

→ robot\_state\_publisher won't publish the transform because [sensor\_messages/JointState](http://docs.ros.org/api/sensor_msgs/html/msg/JointState.html) doesn't have enough information to publish the transform

Lanch order:

<!-- send ur5 urdf to param server →

<!--Publish robot state to TF-->

<!-- driver -->

<node name="ur\_driver" pkg="ur\_modern\_driver" type="ur\_driver" output="screen">

11. Install photoneo driver and photoneo\_camera ROS package

sudo ./PhotoneoPhoXiControlInstaller-1.2.3-Ubuntu16-STAB.run # use this specific version to be compatible with photoneo ros package

modify /opt/PhotoneoPhoXiControl/PhoXiConfig.cmake: gcc version to 5.5.0

cd ~/catkin\_ws/src

git clone https://github.com/as-wanfang/phoxi\_camera.git

cd ..

catkin build

source devel/setup.bash

roslaunch phoxi\_camera phoxi\_camera.launch

基于photoneo的手眼标定代码放在src/cal\_dev/calibration\_ransac\_circle\_zhaodata.py

测试代码src/calibration\_test\_5points.py

12. Modify moveit config for photoneo

/ur5\_moveit\_config/config/sensor\_photoneo\_pointcloud.yaml 设置Octmap的来源：点云或深度图

/ur5\_moveit\_config/launch/sensor\_manager.launch.xml 设置Octmap的性质：

How to trigger photoneo through python:

python

import rospy

from phoxi\_camera.srv import \*

rospy.ServiceProxy('phoxi\_camera/get\_frame', GetFrame)(-1)

Run photoneo demo without a real robot

roslaunch ur5\_moveit\_config demo\_.launch

Run photoneo demo with real robot

13. Install Object Recognition Kichen

http://wg-perception.github.io/object\_recognition\_core/install.html#install

sudo apt-get install ros-kinetic-object-recognition-core

sudo apt-get install ros-kinetic-openni\*

sudo apt-get install libsdl1.2-dev

sudo apt-get install libosmesa6-dev

cd ~/catkin\_ws/src

git clone http://github.com/wg-perception/object\_recognition\_msgs

git clone http://github.com/wg-perception/object\_recognition\_ros

git clone http://github.com/wg-perception/object\_recognition\_ros\_visualization

git clone http://github.com/wg-perception/capture

git clone http://github.com/wg-perception/linemod

git clone <http://github.com/wg-perception/ork_renderer>

14. Install realsense

Install sdk [https://realsense.intel.com/sdk-2/#install](https://realsense.intel.com/sdk-2/" \l "install)

git clone <https://github.com/intel-ros/realsense.git>

catkin build

15. Install Aubo

sudo apt-get install ros-kinetic-industrial-core

git clone <https://github.com/as-wanfang/aubo_robot.git>

roslaunch aubo\_i5\_moveit\_config moveit\_planning\_execution.launch sim:=false robot\_ip:=192.168.1.101