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

方法二

sudo add-apt-repository ppa:graphics-drivers/ppa

sudo apt update

sudo apt install nvidia-390

Ubuntu实时系统不支持Nvidia显卡驱动，因此需要设置intel集显

切换到非实时系统时，需切换回Nvidia显卡

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

nvcc --version

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

E tensorflow/stream\_executor/cuda/cuda\_dnn.cc:363] Loaded runtime CuDNN library: 7.0.5 but source was compiled with: 7.1.4. CuDNN library major and minor version needs to match or have higher minor version in case of CuDNN 7.0 or later version.

sudo apt-get install libcudnn7

libcudnn7 is already the newest version (7.4.1.5-1+cuda9.0).

Download cudnn

sudo dpkg -i libcudnn7\_7.1.4.18-1+cuda9.0\_amd64.deb

sudo dpkg -i libcudnn7-dev\_7.1.4.18-1+cuda9.0\_amd64.deb

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

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 -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==1.12

*failed call to cuInit: CUDA\_ERROR\_UNKNOWN*

### **sudo apt-get install nvidia-modprobe**

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

<http://wiki.ros.org/kinetic/Installation/Ubuntu>

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

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

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

Issue: python: /build/pcl-6\_P28C/pcl-1.7.2/io/src/ply\_io.cpp:500: void pcl::PLYReader::objInfoCallback(const string&): Assertion `st.size () == 3' failed. When read ply files save from photoneo

remove old pcl version 1.7.2:

sudo apt-get remove libpcl\*

The following packages will be REMOVED:

libpcl-apps1.7 libpcl-common1.7 libpcl-conversions-dev libpcl-dev libpcl-doc libpcl-features1.7 libpcl-filters1.7 libpcl-io1.7 libpcl-kdtree1.7 libpcl-keypoints1.7 libpcl-msgs-dev

libpcl-octree1.7 libpcl-outofcore1.7 libpcl-people1.7 libpcl-recognition1.7 libpcl-registration1.7 libpcl-sample-consensus1.7 libpcl-search1.7 libpcl-segmentation1.7

libpcl-surface1.7 libpcl-tracking1.7 libpcl-visualization1.7 libpcl1 libpcl1-dev libpcl1.7 libpcl1.7-dbg ros-kinetic-desktop-full ros-kinetic-pcl-conversions ros-kinetic-pcl-ros

ros-kinetic-perception ros-kinetic-perception-pcl

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

PCL from master, 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:

cd ~

git clone https://github.com/strawlab/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 and python-pcl/build and rebuild

Issue: <https://github.com/udacity/RoboND-Perception-Exercises/issues/18>

modify the following files:

python-pcl/pcl/pxi/PointCloud\_PointXYZ.pxi

python-pcl/pcl/pxi/PointCloud\_PointXYZ\_180.pxi

python-pcl/pcl/pxi/Filters/StatisticalOutlierRemovalFilter.pxi  
python-pcl/pcl/pxi/Filters/StatisticalOutlierRemovalFilter\_180.pxi

cd python-pcl

remove build info under /python-pcl

python setup.py build

sudo python setup.py install

The above modification is hosted on <https://github.com/as-wanfang/python-pcl>

Please git clone from it instead of <https://github.com/strawlab/python-pcl.git>

7. Install dexnet relevant python packages

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 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 .

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

cd /catkin\_ws/

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

8. Install universal\_robot and ur modern driver

cd /catkin\_ws/src/

git clone https://github.com/ros-industrial/universal\_robot.git

cd universal\_robot

rm -rf ur\_driver

git clone <https://github.com/iron-ox/ur_modern_driver.git>

cd ur\_modern\_driver/

git checkout iron-kinetic-devel

add

<!-- Connect tool0\_controller to base using floating joint -->

<link name="tool0\_controller"/>

<joint name="base\_tool0\_controller\_floating\_joint" type="floating">

<origin xyz="0 0 0" rpy="0 0 0"/>

<parent link="${prefix}base"/>

<child link="tool0\_controller"/>

</joint>

to the ~/catkin\_ws/src/universal\_robot/ur\_description/urdf/ur5.urdf.xacro

if use source installed PCL, modify ~/catkin\_ws/src/universal\_robot/ur5\_moveit\_config/CmakeLists.txt:

## Find source installed PCL, comment this if use apt installed pcl

find\_package(PCL 1.8.1 REQUIRED)

INCLUDE\_DIRECTORIES(${PCL\_INCLUDE\_DIRS})

LINK\_DIRECTORIES(${PCL\_LIBRARY\_DIRS})

ADD\_DEFINITIONS(${PCL\_DEFINITIONS})

message("PCL\_LIBRARY\_DIRS" ${PCL\_LIBRARY\_DIRS})

cd ~/catkin\_ws/

catkin build

source devel/setup.bash

Actual robot:

roslaunch ur\_modern\_driver ur5\_bringup.launch limited:=true robot\_ip:=192.168.1.103 [reverse\_port:=REVERSE\_PORT]

Simulation in Gazebo

roslaunch ur\_gazebo ur5.launch

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

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">

9. 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/photoneo/phoxi_camera.git>

cd phoxi\_camera/

git checkout tags/2.0.0

cd ../..

catkin build

source devel/setup.bash

roslaunch phoxi\_camera phoxi\_camera.launch

Issues: The urdf file of photoneo is wrong, changed the Phoxi3Dscanner\_values.xacro line 14:

<origin xyz="${-0.23 / 2.0} 0.0168 0.035" rpy="0 -0.26965337 ${pi}" />

10. Modify UR5\_moveit\_config for photoneo and RG2

/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)

Install rg2 gripper from <https://github.com/sharathrjtr/ur10_rg2_ros>

add rg2 and photoneo to /ur\_description/urdf/ur5\_joint\_limited\_robot.urdf.xacro

Run photoneo demo without a real robot

roslaunch ur5\_moveit\_config demo\_.launch

Run photoneo demo with real robot

11. Install Object Recognition Kichen

with source installed pcl1.8.1

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

with pcl1.7 installed by apt-get

sudo apt-get install ros-kinetic-object-recognition-\*

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

sudo apt-get install libsdl1.2-dev

sudo apt-get install libosmesa6-dev

12. 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>

cd realsense

git checkout tag/2.1.0

catkin build

roslaunch realsense2\_camera rs\_rgbd.launch

rosrun rviz rviz

13. Aubo

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

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

**rpy** (optional: defaults **'to zero vector**'if not specified)

* + Represents the rotation around fixed axis: first roll around x, then pitch around y and finally yaw around z. All angles are specified in radians.

roslaunch aubo\_i5\_moveit\_config demo.launch

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

13. Franka-ros

cd ~

git clone --recursive https://github.com/frankaemika/libfranka

cd libfranka

mkdir build

cd build

cmake -DCMAKE\_BUILD\_TYPE**=**Release ..

cmake --build .

git clone --recursive https://github.com/frankaemika/franka\_ros src/franka\_ros

catkin build -DFranka\_DIR:PATH=/home/bionicdl/libfranka/build

Gazebo仿真:

1. /franka\_description/robots/panda\_arm.urdf.xacro中给个关节添加驱动transmission

2. panda\_arm\_hand.urdf.xacro/common.gazebo.xacro中添加libgazebo\_ros\_control.so，这是controller manager,其中namespace要与controller\_utils.lanunch中spawner’s namespace一致，否则步骤3无法启动controller，本仿真中没有使用namespace。

3. 添加控制器配置文件franka\_gazebo/controller/controller\_franka.yaml，

<rosparam file="$(find ur\_gazebo)/controller/arm\_controller\_ur5.yaml" command="load"/>

<node name="arm\_controller\_spawner" pkg="controller\_manager" type="spawner" args="arm\_controller gripper\_controller" respawn="false" output="screen"/>

成功启动controller后在rostopic list里会有/arm\_controller/follow\_joint\_trajectory/

4. panda\_moveit\_config中，config的controllers\_gazebo.yaml配置，name要与franka\_gazebo/controller/controller\_franka.yaml中的namespace保持一直，使用“arm\_controller”和“gripper\_controller”.

5. 当加载.world文件时，请仔细检查文件中的模型文件所在的目录添加到了GAZEBO\_MODEL\_PATH路径中，否则会导致gazebo启动异常缓慢无法加载世界。加载世界和spawn机器人可在upload urdf后进行，而后再spawn controllers

roslaunch franka\_gazebo franka.launch包含

roslaunch panda\_moveit\_config panda\_moveit\_planning\_execution.launch sim:=true

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

controllers:

joint\_state\_controller/JointStateController

position\_controllers/JointPositionController world\_joint

position\_controllers/JointTrajectoryController 手臂

effort\_controllers/JointTrajectoryController 平行电爪

添加依赖的ros包后需要在package.xml文件中添加run\_depeng。

添加新的srv或src文件，需要在CMakeList.txt文件中添加对应部分

加载world后机械臂会垮掉，将机械臂、相机、抓取物品放在同一个world文件里。

解决：将panda\_arm.xacro里机械臂的joint的safety\_controller注释掉。加进的物体无法抓取的原因是gazebo默认的model是static，需要设置为false。

sudo apt-get install \

ros-kinetic-gazebo-ros \

ros-kinetic-eigen-conversions \

ros-kinetic-object-recognition-msgs \

ros-kinetic-roslint

sudo apt-get install libopenni2-dev libopenni2-0

remove python-opengl folder directly in /usr/lib/python2.7/dist-packages

用gqcnn和franka做抓取

cd catkin\_ws/src/gqcnn

git checkout 2.0

cd ~/catkin\_ws

catkin clean gqcnn

catkin build gqcnn

roslaunch franka\_gazebo franka.launch

roslaunch gqcnn gqcnn.launch

rosrun franka\_description project\_grasp.py

14. Install point\_cloud\_io

cd ~/catkin\_workspace/src

git clone https://github.com/anybotics/point\_cloud\_io.git

cd ../

catkin build

rosrun point\_cloud\_io read \_file\_path:=/home/bionicdl/photoneo\_data/20181217/waypoint5.ply \_topic:=/phoxi\_camera/pointcloud \_frame:=/PhoXi3Dscanner\_sensor \_rate:=1.0

rosrun point\_cloud\_io write \_topic:=/your\_topic \_folder\_path:=/home/user/my\_point\_clouds

14. Install checkerboard\_detector

sudo apt-get install ros-kinetic-jsk-recognition-msgs

ros-kinetic-posedetection-msgs

sudo apt-get install ros-kinetic-dynamic-tf-publisher

rosrun point\_cloud\_io read \_file\_path:=flange\_ground\_truth/wrist1.ply \_topic:=/phoxi\_camera/pointcloud \_frame:=/wrist\_1\_link \_rate:=1.0

rosrun point\_cloud\_io read \_file\_path:=YCB/YCB\_items\_high\_0000.ply \_topic:=/phoxi\_camera/pointcloud \_frame:=/motionX \_rate:=1.0