https://github.com/jobbole/awesome-programming-books

1.安装搜狗 http://jingyan.baidu.com/article/08b6a591cb06f114a8092209.html

2.安装lantern

<https://github.com/getlantern/lantern>

最后推荐用安装deb文件，下载deb后

sudo dpkg -i lantern-installer-beta-64-bit.deb

卸载：sudo apt-get autoremove --purge lantern

搜索中输入lantern，点开

3.安装ros indigo

http://www.jianshu.com/p/04be841e2293

最后$echo “source ~/catkin\_ws/devel/setup.bash” >> ~/.bashrc

在其他地方增加包路径：比如，.bashrc中添加

export ROS\_PACKAGE\_PATH=${ROS\_PACKAGE\_PATH}:~/ORB\_SLAM2/Examples/ROS

编译：

mkdir build

cd build

cmake .. -DROS\_BUILD\_TYPE=Release

make -j4

4.安装openni2 nite2 for(asus xtion)

http://stackoverflow.com/questions/19938830/openni2-and-asus-xtion-pro-live-device-not-found

sudo apt-get install g++ python libusb-1.0-0-dev libudev-dev openjdk-6-jdk freeglut3-dev doxygen graphviz -y

git clone https://github.com/occipital/OpenNI2.git

cd OpenNI2 && make -j4

export OPENNI2\_INCLUDE=/home/keetsky/OpenNI2/Include

sudo ln -s /lib/x86\_64-linux-gnu/libudev.so.1.3.5 /lib/x86\_64-linux- gnu/libudev.so.0

运行./NiViewer

但是发现无法编译ros含有openni.h的tracker\_my的程序，

安装OpenNI-Linux-x64-2.2

sudo ./install.sh

拷贝OpenNIDevEnvironment到.bashrc中

sudo cp -r Include /usr/include/openni2

sudo cp -r Redist/OpenNI2 /usr/lib

sudo cp Redist/libopenNI2.\* /usr/lib

sudo ln -s /lib/x86\_64-linux-gnu/libudev.so.1.3.5 /lib/x86\_64-linux- gnu/libudev.so.0

安装Sensor-Bin-Linux-x64-v5.1.6.6：

chmod +x install.sh && sudo ./install.sh

安装nite2:(NiTE-2.0.0)

./install.sh

拷贝NiTEDevEnvironment文件中的

export NITE2\_INCLUDE=/home/keetsky/NiTE-2.0.0/Include

export NITE2\_REDIST64=/home/keetsky/NiTE-2.0.0/Redist

到 .bashrc 中

运行./UserV iewer

5.boost 安装

sudo apt-get install libboost-all-dev

或sudo apt-get install libboost-dev libboost-thread-dev libboost-filesystem-dev

6.eigen3安装

sudo apt-get install libeigen3-dev

或者：下载http://eigen.tuxfamily.org/index.php?title=Main\_Page

mkdir build

cd build

cmake ..

make

sudo make install

7.ros usb\_camera 安装

cd catkin\_ws/src

git clone https://github.com/bosch-ros-pkg/usb\_cam

cd ..

catkin\_make

roscore

roslaunch usb\_cam usb\_cam-test.launch

法2：

sudo apt-get install ros-indigo-uvc-camera

sudo apt-get install ros-indigo-usb-cam

8.ros openni2 安装

http://blog.csdn.net/chenxingwangzi/article/details/48825767

http://www.cnblogs.com/cj2014/p/4007165.html

sudo apt-get install ros-indigo-openni2-camera

sudo apt-get install ros-indigo-openni2-launch

roslaunch openni2\_launch openni2.launch

rosrun rviz rviz

或：rosrun rqt\_image\_view rqt\_image\_view

9.kinect2 驱动安装

http://blog.csdn.net/sunbibei/article/details/51594824

git clone https://github.com/OpenKinect/libfreenect2.git

cd libfreenect2

cd depends; ./download\_debs\_trusty.sh

sudo apt-get install build-essential cmake pkg-config

sudo dpkg -i debs/libusb\*deb

sudo apt-get install libturbojpeg libjpeg-turbo8-dev

sudo dpkg -i debs/libglfw3\*deb

sudo apt-get install -f

sudo apt-get install libgl1-mesa-dri-lts-vivid 出现错误可忽略

mkdir build && cd build

cmake .. -DCMAKE\_INSTALL\_PREFIX=$HOME/freenect2 -DENABLE\_CXX11=ON

make

make install

sudo cp ../platform/linux/udev/90-kinect2.rules /etc/udev/rules.d/

查看运行结果：./bin/Protonect

10.orb slam2 安装

http://webdiis.unizar.es/~raulmur/orbslam/

https://github.com/raulmur/ORB\_SLAM2

http://www.liuxiao.org/2016/03/ros-orb-slam2-%E5%AE%89%E8%A3%85%E8%B0%83%E8%AF%95/

http://blog.csdn.net/fk1174/article/details/51800728

http://blog.csdn.net/u010128736/article/category/6461394

先安装依赖项

Pangolin：

sudo apt-get install libglew-dev

git clone https://github.com/stevenlovegrove/Pangolin.git

cd Pangolin

mkdir build

cd build

cmake ..

make -j

opencv:

eigen3:

安装：

git clone https://github.com/raulmur/ORB\_SLAM2.git ORB\_SLAM2

cd ORB\_SLAM2

chmod +x build.sh

./build.sh

下载数据库解压：http://vision.in.tum.de/data/datasets/rgbd-dataset/download

比如：解压文件夹为rgbd\_dataset\_freiburg1\_xyz

运行：./Examples/Monocular/mono\_tum Vocabulary/ORBvoc.txt Examples/Monocular/TUM1.yaml /home/keetsky/ORB\_SLAM2/rgbd\_dataset\_freiburg1\_xyz

将ORBvoc.txt替换为ORBvoc.bin文件说是可提升载入速度或运行速度，未测试：https://github.com/sakurazhu/ORB\_SLAM2-1/blob/master/Vocabulary/ORBvoc.bin

运行KITTI：./Examples/Monocular/mono\_kitti Vocabulary/ORBvoc.txt Examples/Monocular/KITTI00-02.yaml /media/keetsky/469EAFDF9EAFC5B1/BaiduYunDownload/download/dataset/sequences/02

ROS:

添加ROS\_PACKAGE\_PATH路径:在.brashrc中添加

export ROS\_PACKAGE\_PATH=${ROS\_PACKAGE\_PATH}:~/ORB\_SLAM2/Examples/ROS

或用命令：$echo “export ROS\_PACKAGE\_PATH=${ROS\_PACKAGE\_PATH}:~/ORB\_SLAM2/Examples/ROS” >> ~/.bashrc

查看是否添加成功： echo $ROS\_PACKAGE\_PATH

用单目摄像头： usb\_camera，运行ros 摄像头 roslaunch usb\_cam usb\_cam-test.launch rostopic list 获得摄像头节点发布主 题/usb\_cam/image\_raw,修改ORB\_SLAM2/Examples/ROS/ORB\_SLAM2/src/ros\_mono.cc中为ros::Subscriber sub = nodeHandler.subscribe("/usb\_cam/image\_raw",

即usb\_camera把摄像头数据发布到topic

编译ros程序

cd ~/ORB\_SLAM2

chmod +x build\_ros.sh

./build\_ros.sh

启动roscore 和usb roslaunch

启动mono：rosrun ORB\_SLAM2 Mono /home/keetsky/ORB\_SLAM2/Vocabulary/ORBvoc.txt /home/keetsky/ORB\_SLAM2/Examples/Monocular/TUM1.yaml 可看到结果

深度数据：采用asus xtion ,需要预先安装openni2 camera 和openni2 launch。运行roslaunch openni2\_launch openni2.launch

修改orb\_slam 中ros\_rgbd.cc程序，发布主题修改为/camera/rgb/image\_raw 和camera/depth\_registered/image\_raw

rosrun ORB\_SLAM2 RGBD /home/keetsky/ORB\_SLAM2/Vocabulary/ORBvoc.txt /home/keetsky/ORB\_SLAM2/Examples/Monocular/TUM1.yaml

深度数据：kinect2 安装完驱动安装 iai-kinect2

http://blog.csdn.net/sunbibei/article/details/51594824

cd ~/catkin\_ws/src/

git clone https://github.com/code-iai/iai\_kinect2.git

cd iai\_kinect2

rosdep install -r --from-paths . （我运行出错，跳过后不影响结果）

cd ~/catkin\_ws

catkin\_make -DCMAKE\_BUILD\_TYPE="Release"

如果路径不对可catkin\_make -Dfreenect2\_DIR=path\_to\_freenect2/lib/cmake/freenect2 -DCMAKE\_BUILD\_TYPE="Release"

roslaunch kinect2\_bridge kinect2\_bridge.launch

查看：rosrun kinect2\_viewer kinect2\_viewer sd cloud

修改ORB\_SLAM2下ros\_rgbd.cc程序

节点主题改为"/kinect2/qhd/image\_color" 和"/kinect2/qhd/image\_depth\_rect"

重新编译：cd ~/ORB\_SLAM2

./build\_ros.sh

运行：roscore

roslaunch kinect2\_bridge kinect2\_bridge.launch

rosrun ORB\_SLAM2 RGBD /home/keetsky/ORB\_SLAM2/Vocabulary/ORBvoc.txt /home/keetsky/ORB\_SLAM2/Examples/Monocular/TUM1.yaml

11.python2.7安装

sudo apt-get install libpython2.7-dev

12.glew 安装

sudo apt-get install libglew-dev

13.cmake 安装

sudo apt-get install cmake

14.git 安装

sudo apt-get install git

15.编译基础库安装

sudo apt-get install build-essential

16.opencv 安装

sudo apt-get install build-essential

sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev

sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev

下载opencv 2.4.11

mkdir release

cd release

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local ..

make

sudo make install

python 版本：sudo apt-get install python-opencv

sudo apt-get install python-numpy

:conda install -c https://conda.binstar.org/menpo opencv

在anaconda中安装：conda install --channel [http://conda.anaconda.org/menpo opencv3](http://conda.anaconda.org/menpo%20opencv3)

conda install –c <https://conda.binstar.org/menpo> opencv3 # 这个源成功安装了

也可在[www.lfd.uci.edu/~gohlke/pythonlibs/](http://www.lfd.uci.edu/~gohlke/pythonlibs/) 下载对应opencv版本.whl文件

pip install opencv\_python-3.2.0-cp35-cp35m-win\_amd64.whl

numpy安装法2：下载http://www.scipy.org/scipylib/download.html

unzip numpy-1.11.1.zip

cd numpy-1.11.1/

python setup.py install

或：pip install . pip install numpy

这里先提示以下这个地方，make的过程中会出现这个文件“cv.so”，它是连接OpenCV与python的关键文件

出现“No module named cv2”，原因就是python找不到我们刚才提到的cv.so文件， 我们进入路径/usr/local/lib/python2.7或**/usr/lib/python2.7**中发现有两个文件夹，一个是dist-package，一个是site- package，python默认只在前者里搜索，而我们生成的cv.so/cv2.so文件则在我们的安装目录"../../opencv2.1.10 /release"中（还记得这个目录吧，就是我们刚才在安装过程中建的那个目录，如果你的安装目录中没有这个文件，则说明前面的make步骤出问题了， 可能需要重新走一遍整个安装过程)，我们只需要简单的将它移动到dist-package文件夹即可。

17.安装BLAS and LAPACK库

sudo apt-get install libblas-dev

sudo apt-get install liblapack-dev

18.显卡驱动安装quadro k620

查看显卡型号：lspci | grep -i nvidia 或sudo lshw -numeric -C display

查看参数ubuntu-drivers devices 获得参数为375

<http://www.nvidia.com/Download/index.aspx?lang=en-us下载显卡驱动>

修改下载包权限比如：sudo chmod +x NVIDIA\*\*.run

关闭X-Window，很简单：sudo service lightdm stop，然后切换到tty1：Ctrl+Alt+F1即可

接下来就是最关键的一步了：sudo ./NVIDIA.run开始安装，安装过程比较快

最后安装完毕后，重新启动X-Window：sudo service lightdm start，然后Ctrl+Alt+F7进入图形界面；

查看是否成功：$nvidia-smi 或$nvidia-settings

卸载：sh ./nvidia.run –uninstall 或sudo apt-get remove --purge nvidia-331-updates sudo apt-get remove --purge nvidia\*

法2：Sudo apt-get update##更新源

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

按回车继续

sudo apt-get update  源不好可 sudo apt-get -qy update  
 sudo apt-get install nvidia-375   
 sudo apt-get install mesa-common-dev   
 sudo apt-get install freeglut3-dev

或：sudo apt-get install nvidia-331 nvidia-settings nvidia-prime

18\_2.笔记本双显卡驱动安装 GTX 1050i

<http://blog.csdn.net/10km/article/details/61191230>

<http://www.cnblogs.com/devilmaycry812839668/p/6867780.html>

1. 设置BIOS:BIOS中Secure Boot设置为disable，否则安装完nvidia驱动后可能进步了桌面，正安装NVIDIA driver后，如果再将Secure Boot设置为enable,再启动ubuntu则NVIDIA driver则不会工作。
2. 禁用nouveau驱动：(Ubuntu系统集成的显卡驱动程序是nouveau，我们需要先将nouveau从[Linux](http://lib.csdn.net/base/linux" \o "Linux知识库" \t "_blank)内核卸载掉才能安装NVIDIA官方驱动。 )查看属性：ll /etc/modprobe.d/blacklist.conf 修改属性：sudo chmod 666 /etc/modprobe.d/blacklist.conf sudo vi /etc/modprobe.d/blacklist.conf文件末尾添加：blacklist vga16fb

blacklist nouveau

blacklist rivafb

blacklist rivatv

blacklist nvidiafb

修改保存，复原文件属性：sudo chmod 644 /etc/modprobe.d/blacklist.conf

1. 更新内核后重启电脑：sudo update-initramfs -u
2. 确认nouveau已经被屏蔽掉(无显示)，否则后面会安装后进不了桌面：lsmod | grep nouveau
3. 安装NVIDIA显卡驱动：sudo add-apt-repository ppa:graphics-drivers/ppa sudo apt-get update 且sudo apt-get upgrade 且sudo apt-get install build-essentia查看可用驱动版本：ubuntu-drivers devices 比如：nvidia-381安装NVIDIA driver:进入tty1文本模式->关闭图形桌面管理器：sudo service lightdm stop ->安装：sudo apt-get install nvidia-381 如果所有都完成后无法进入桌面可进入tty1卸载显卡驱动换一个驱动版本安装试下 ->重启：sudo reboot
4. 查看驱动是否成功：sudo nvidia-smi 和 sudo nvidia-settings
5. 以上步骤尽量严格完成，否则很可能显卡安装失败或者无法进入登入界面或桌面系统

19.CUDA8.0安装

要求先安装完显卡驱动

下载cuda8.0:https://developer.nvidia.com/cuda-release-candidate-download

安装：sudo sh cuda\_8.0.27\_linux.run

在选择Install NVIDIA Accelerated Graphics Driver for Linux-x86\_64 361.62?  
(y)es/(n)o/(q)uit: n 其他选择YES

如果安装失败可能需要安装缺少依赖库：sudo apt-get install freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libgl1-mesa-glx libglu1-mesa libglu1-mesa-dev

设置环境变量(最好将其写在ros环境变量前否则可能导致roscore无法运行或运行出错).bashrc：$export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}

export LD\_LIBRARY\_PATH=/usr/local/cuda-8.0/lib64\${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

动态链接库:$ sudo gedit /etc/profile 文件尾部填入export PATH = /usr/local/cuda/bin:$PATH

创建链接文件：sudo gedit /etc/ld.so.conf.d/cuda.conf 填入：/usr/local/cuda/lib64

然后执行：sudo ldconfig

验证安装是否完成：nvidia-smi

测试：cd /usr/local/cuda-8.0/samples/1\_Utilities/deviceQuery

sudo make

./deviceQuery

20.cudnn 安装

下载完cudnn5.1之后进行解压，cd进入cudnn5.1,tar xvzf cudnn-8.0-linux-x64-v4.tgz

解压之后的include目录，在命令行进行如下操作：

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

sudo chmod a+r /usr/local/cuda/lib64/libcudnn\*

或：sudo cp cudnn.h /usr/local/cuda/include/ #复制头文件

sudo cp lib\* /usr/local/cuda/lib64/ #复制动态链接库

cd /usr/local/cuda/lib64/

sudo rm -rf libcudnn.so libcudnn.so.5 #删除原有动态文件

sudo ln -s libcudnn.so.5.0.5 libcudnn.so.5 #生成软衔接

sudo ln -s libcudnn.so.5 libcudnn.so #生成软链接

sudo ldconfig #更新链接库

如果存在运行opencv程序发送链接cuda opencv相关错误可在程序cmakelists.txt设置：

set(CUDA\_USE\_STATIC\_CUDA\_RUNTIME OFF)

或对生成的CMakeCache.txt修改为CUDA\_RUNTIME ：OFF

20.vim 配置

<https://github.com/sakurazhu/vim>

21.qq安装

安装之前，假设你已经安装好了wine，如果没有，搜索ubuntu14.04 wine就有很多教程:   
sudo add-apt-repository ppa:ubuntu-wine/ppa   
sudo apt-get update   
sudo apt-get install wine1.7 winetricks

我的总结:

1. 在上面链接中下载WineQQ7.8-20151109-Longene.deb（http://www.longene.org/forum/viewtopic.php?f=6&t=30516）
2. 如果之前安装了其他了版本qq，可以先卸载：   
   step1.找到安装版本qq的文件夹，比如/opt/longene/qq   
   step2.打开里面的qq.sh文件，查看你安装版本的qq名package\_name = wine-qq7.8-longeneteam   
   step3.卸载sudo dpkg –purge wine-qq7.8-longeneteam
3. 安装：   
   32bit系统直接安装: sudo dpkg -i /home/hyj/WineQQ7.8-20151109-Longene.deb 。   
   64bit, 先安装32位兼容库sudo apt-get install libgtk2.0-0:i386，再使用上面的安装指令。

（转载请注明作者和出处：<http://blog.csdn.net/heyijia0327> 未经允许请勿用于商业用途）

22.串口通讯：

<http://blog.csdn.net/zhaodan19861107/article/details/7766338>

查看串口名称使用 [ls](https://www.baidu.com/s?wd=ls&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLuWIhryD4PHmdnyFWuWcz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjD4nHf3PHR4) [-l](https://www.baidu.com/s?wd=-l&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLuWIhryD4PHmdnyFWuWcz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjD4nHf3PHR4) /dev/ttyS\*

查看串口设备：dmesg | grep ttyS\* 可获得可用串口和波特率

在Linux下如果出现Cannot open /dev/ttyS0: Permission denied.普通用户 没有权限读写串口设备。inux下的串口设备名通常是/dev/ttyS[0-9]或 者/dev/ttyUSB[0-9]，后者对应的是USB转串口的设备。

$ sudo vim /etc/udev/rules.d/90-serials.rules 文件中写入

KERNEL=="ttyS[0-9]",NAME="%k",GROUP="tty",MODE="0666"

KERNEL=="ttyUSB[0-9]",NAME="%k",GROUP="tty",MODE="0666"

重启电脑后即可

23. vim 配置 1

https://github.com/amix/vimrc

<https://github.com/sakurazhu/vim>

<https://github.com/ma6174/vim>

wget -qO- https://raw.github.com/ma6174/vim/master/setup.sh | sh -x

vim配置2：

使用ctags可以在变量之间跳跃。执行：apt-get install ctags

apt-get install vim-scripts vim-addons install taglist

安装pydiction用来实现代码补全和语法提示功能：在 http://www.vim.org/scripts/script.php?script\_id=850下载，将包下$cp after/ftplugin/python\_pydiction.vim ~/.vim/after/ftplugin（如 果/after/ftplugin不存在，则创建）,cp complete-dict ~/.vim, cp pydiction.py ~/.vim

编辑～/.vimrc 文件添加：

let Tlist\_Auto\_Highlight\_Tag=1

let Tlist\_Auto\_Open=1

let Tlist\_Auto\_Update=1

let Tlist\_Display\_Tag\_Scope=1

let Tlist\_Exit\_OnlyWindow=1

let Tlist\_Enable\_Dold\_Column=1

let Tlist\_File\_Fold\_Auto\_Close=1

let Tlist\_Show\_One\_File=1

let Tlist\_Use\_Right\_Window=1

let Tlist\_Use\_SingleClick=1 nnoremap <silent> <F8> :TlistToggle<CR> filetype plugin on

autocmd FileType python

set omnifunc=pythoncomplete#Complete

autocmd FileType javascrīpt set omnifunc=javascriptcomplete#CompleteJS

autocmd FileType html set omnifunc=htmlcomplete#CompleteTags

autocmd FileType css set omnifunc=csscomplete#CompleteCSS

autocmd FileType xml set omnifunc=xmlcomplete#CompleteTags

autocmd FileType php set omnifunc=phpcomplete#CompletePHP

autocmd FileType c set omnifunc=ccomplete#Complete

let g:pydiction\_location='~/.vim/tools/pydiction/complete-dict'

set autoindent set tabstop=4

set shiftwidth=4

set expandtab

set numbe

24. python 环境搭建

<http://blog.csdn.net/kingppy/article/details/13080919>

bash-it可以美化你的 bash 环境，让你更高效地使用控制台终端

安装第三方库如：sudo apt-get install python-numpy python-scipy python-matplotlib python-sklearn ipython ipython-notebook python-pandas 采用ipython 编辑器输入ipython

或用pip:sudo pip install numpy

sudo pip install scipy

**安装anaconda:** [**https://mirrors.tuna.tsinghua.edu.cn/help/anaconda/**](https://mirrors.tuna.tsinghua.edu.cn/help/anaconda/)**下载对应**

bash Anaconda3-242-linux-x86\_64.sh

都yes

source ~/.bashrc

conda list 如果没反应说明配置不对，则手动配置环境

sudo gedit ~/.bashrc

export PATH=”/home/keetsky/anaconda3/bin:$PATH”

终端输入python –V 则显示为anaconda3种python 的版本

终端输入spyder 打开编译器

多版本切换：

<http://blog.csdn.net/lixintong1992/article/details/51568474> ([Python多版本切换工具-Pyenv\virtualenv及Anaconda科学计算环境的配置](http://blog.csdn.net/lixintong1992/article/details/51568474) ) :

(1)安装pyenv:$ git clone https://github.com/yyuu/pyenv.git ~/.pyenv

$echo 'export PYENV\_ROOT="$HOME/.pyenv"' >> ~/.bashrc

$ echo 'export PATH="$PYENV\_ROOT/bin:$PATH"' >> ~/.bashrc

$ echo 'eval "$(pyenv init -)"' >> ~/.bashrc

source ~/.bashrc

(2)查看pyenv可安装版本列表,$ pyenv install –list ,安装制定的版本

如$ pyenv install 3.5.2 安装在~.pyenv/versions文件夹下。使用pip安装的包完成之后，可能需要对**[数据库](http://lib.csdn.net/base/mysql)**进行更新:pyenv rehash 。卸载指定的python版本

：pyenv uninstall anaconda3-4.2.0 。查看当前已经安装了的python版本：pyenv versions，system是指系统的python版本；\*表示当前环境所处于的python版本。如anaconda之类大容量的版本，由于网络的问题，总是连接中断，安装失败。此时可以先从官方网站下载安装包，然后放在~/.pyenv/cache文件夹中，然后在pyenv install anaconda3-4.3.0此版本，pyenv会自动先从此文件夹中搜索,最好安装在.pyenv/version中。全局版本切换：pyenv global anaconda-2.4.0，局部版本切换：pyenv local python3.4.1，切换会系统版本pyenv global system。输入conda list 就可以查询，你现在安装了哪些库  
conda list；conda install \*\*安装库如conda install scipy；在终端输入spyder开启IDE;更新conda update conda，conda update anaconda；

SPYDER: Installing: sudo pip install spyder

* Updating: sudo pip install -U spyder

（3）virtualenv创建纯净虚拟环境

git clone https://github.com/yyuu/pyenv-virtualenv.git ~/.pyenv/plugins/pyenv-virtualenv

echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.bash\_profile

source ~/.bash\_profile

创建一个2.7.1的虚拟环境：pyenv virtualenv 2.7.1 env271 ，这个环境的真实目录位于：~/.pyenv/versions/ ；注意，命令中的 ‘2.7.1’ 必须是一个安装前面步骤已经安装好的python版本， 否则会出错。然后我们可以继续通过 ‘pyenv versions’ 命令来查看当前的虚拟环境。

## 切换和使用新的python虚拟环境：pyenv activate env271 ；如果要切换回系统环境， 运行这个命令即可pyenv deactivate ；

## 删除这个虚拟环境：rm -rf ~/.pyenv/versions/env271/或pyenv uninstall env271

# （4）Anaconda科学计算包的使用：查看anaconda安装自带的包：使用conda list命令的环境时python版本切换到anaconda版本下。给anaconda安装包：conda install ××××

如果要[package-name]=x.x #指定包的版本。

<http://www.cnblogs.com/zangyu/p/5574551.html>

查看已安装版本：$ ls /usr/bin/python\* 结果比如/usr/bin/python /usr/bin/python2 /usr/bin/python2.7 /usr/bin/python3 /usr/bin/python3.4 /usr/bin/python3.4m /usr/bin/python3m

查看现在使用的版本：$ python –version 如Python 2.7.8

法1：打开该用户的 ~/.bashrc 文件，写入alias python='/usr/bin/python3.4'

$~/.bashr 后再查看当前使用版本变为Python 3.4.2

法2：忽略

喜欢的最终环境：需要安装pyenv , pyenv 安装anaconda\*\*\*\*\*，并版本globel切换到anaconda , 查看which python && spyder 目录，打开IDE $spyder。切换回系统版本pyenv globel system

25.bantu时间校准

http://xiaxianpa.blog.163.com/blog/static/2415590622014927102534497/

sudo ntpdate ntp.ubuntu.com

26.V4L usbcamera

查看摄像头是否支持UCV：lsusb 获得：Bus 001 Device 004: ID 05a3:9230 ARC International

lsusb -t

dmesg | grep Cam

lsusb -d 05a3:9230 -v | grep "14 Video" 显示有l4Video 则表示支持

使用camorama查看： 输入命令：sudo apt-get install camorama  
 安装完成后，在终端中输入命令：camorama，即可显示出视频信息；或  
 在菜单“应用程序”中可以打开程序。

27.tensorflow 安装:

anaconda search –t conda tensorflow #查看别人上传版本信息

anaconda show jdfdj/tensorflow 在按提示安装

<http://blog.csdn.net/zhaoyu106/article/details/52793183/> <http://blog.csdn.net/u014365862/article/details/53868411>

windows:http://blog.csdn.net/m0\_37568067/article/details/60468746

anaconda:<http://www.cnblogs.com/willnote/p/6746499.html> http://www.cnblogs.com/aloiswei/p/6510355.html

先安装完CUDA和cudnn(需要保证安装正确否则TensorFlow运行会出错)

运行github tensorflow下载文件sudo pip install tensorflow-0.8.0-cp27-none-linux\_x86\_64.whl

import tensorflow

tensorflow.\_\_version\_\_

输出版本为：'0.11.0rc1'

或：>>> import tensorflow as tf

>>> hello = tf.constant('Hello, TensorFlow!')

>>> sess = tf.Session()

>>> print(sess.run(hello))

Anaconda3-4.2.0 中安装TensorFlow：（假定安装完anaconda）查看官网教程并添加些东西（以下为中文字符写的实际输入可能需要手动输入）：

conda create –n tensorflow python 或指定版本python=3.5

source activate tensorflow #如果是windows 下为：activate tensorflow

也可不按照在虚拟环境中，而直接安装在当前环境中则上面步骤可不执行，

pip install –-ignore-installed –-upgrade https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\_gpu-1.1.0-cp35-cp35m-linux\_x86\_64.whl #GPU版本,python3.5版本

如果下不动：可先到github下载对应 .whl 文件并修改权限放到 anaconda3/lib/python3.5/site-packages或在当前目录下运行：

pip install --ignore-installed --upgrade tensorflow\_gpu-1.1.0-cp35-cp35m-linux\_x86\_64.whl

如果失败则anaconda3要改权限 sudo chmod 777 –R anaconda3 或者升级下 python –m pip install –upgrade pip