OpenCV 3.1 Installation with Python 2.7 and ROS Indigo on Ubuntu 14.04

OpenCV 3.1 Installation

Simply ctrl + c and shift+ctrl + v(Remember to change some path based on your own computer)

```
sudo apt-get -y update
sudo apt-get -y upgrade
sudo apt-get -y dist-upgrade
sudo apt-get -y autoremove


sudo apt-get install build-essential git cmake pkg-config
sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev
sudo apt-get install libilmbase-dev libilmbase6 libopenexr6
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev
sudo apt-get install libv4l-dev libjpeg-dev libopenexr-dev
sudo apt-get install libgtk2.0-dev libeigen3-dev
sudo apt-get install libatlas-base-dev gfortran  libtbb-dev qt5-default libvtk6-dev
sudo apt-get install -y libdc1394-22-dev libavcodec-dev libavformat-dev
sudo apt-get install libswscale-dev libtheora-dev
sudo apt-get install libvorbis-dev libxvidcore-dev libx264-dev
sudo apt-get install yasm libopencore-amrnb-dev
sudo apt-get install libopencore-amrwb-dev libv4l-dev libxine2-dev
sudo apt-get install -y doxygen
sudo apt-get install -y python-dev python-tk python-numpy
```

```
cd ~/<my_working_directory>
git clone https://github.com/Itseez/opencv.git
git clone https://github.com/Itseez/opencv_contrib.git
cd opencv
mkdir build
cd build
```

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D OPENCV_EXTRA_MODULES_PATH=/home/chenta
```

Note: Please change the `OPENCV_EXTRA_MODULES_PATH` according to your own environment And you may not be able to finish this step because it failed to download `ippicv_linux_20151201.tgz`. To solve this, you can manually download `ippicv_linux_20151201.tgz` (https://raw.githubusercontent.com/Itseez/opencv_3rdparty/81a676001ca8075ada498583e4166079e5744668/ippicv/ippicv_linux_20151201.tgz), and put it under `.../opencv/opencv/3rdparty/ippicv/downloads/linux-808b791a6eac9ed78d32a7666804320e`.

```
make -j8
sudo make install
```

If you find that cv2.so was not generated after installation, you may need to comment the `export PATH="/home/chentao/software/anaconda2/bin:$PATH"` in ~/.bashrc to use the python in /usr/bin/python

```
sudo gedit /etc/ld.so.conf.d/opencv.conf
```

Add the following line at the end of the file(it may be an empty file, that is ok) and then save it:

```
/usr/local/lib
```

Run the following code to configure the library:

```
sudo ldconfig
```

Then,

```
sudo gedit /etc/bash.bashrc
```

Add these two lines at the end of the file and save it:

```
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig
export PKG_CONFIG_PATH
```

If you have anaconda installed in your computer, you might need to create a symbolic link for cv2.so so as to let the python in anaconda find this file. Here is an example(you might need to change the directory according to where you install anaconda).

```
cd ~/software/anaconda2/lib/python2.7/site-packages
sudo ln -s /usr/local/lib/python2.7/dist-packages/cv2.so cv2.so
```

To work with ROS Indigo

To recompile cv_bridge so as to make it fit with OpenCV 3.1, do the following commands:

```
mkdir -p ~/catkin_cv/src
cd ~/catkin_cv/src
catkin_init_workspace
cd ~/catkin_cv/
catkin_make
gedit ~/.bashrc
```

Modify ~/.bashrc to make it contain these lines (order matters here):

```
source /opt/ros/indigo/setup.bash
source /home/chentao/catkin_ws/devel/setup.bash
source /home/chentao/catkin_cv/devel/setup.bash
```

```
cd ~/catkin_cv
git clone https://github.com/ros-perception/vision_opencv.git
git clone  https://github.com/ros-perception/image_pipeline.git
```

`image_pipeline` is optional. You need to recompile the packages in `image_pipeline` if you want to use them with the recompiled cv_bridge.

Copy the `cv_bridge` in the downloaded `vision_opencv` folder and all the packages except `camera_calibration` in `image_pipeline` (optional) to ~/catkin_cv_src.

```
cd ~/catkin_cv
catkin_make
```