(\* the following procedures must be strictly followed \*)（\*下列过程必现严格执行\*）

#### Switch source list source

1. Edit sudo VIM / etc / apt/ source.list , replace the relevant source content as follows(编辑sudo vim /etc/apt/source.list，替换相关的源内容，相关内容如下：)

*deb http://mirrors.aliyun.com/ubuntu-ports/ xenial main*

*deb-src http://mirrors.aliyun.com/ubuntu-ports/ xenial main*

*deb http://mirrors.aliyun.com/ubuntu-ports/ xenial-updates main*

*deb-src http://mirrors.aliyun.com/ubuntu-ports/ xenial-updates main*

*deb http://mirrors.aliyun.com/ubuntu-ports/ xenial universe*

*deb-src http://mirrors.aliyun.com/ubuntu-ports/ xenial universe*

*deb http://mirrors.aliyun.com/ubuntu-ports/ xenial-updates universe*

*deb-src http://mirrors.aliyun.com/ubuntu-ports/ xenial-updates universe*

*deb http://mirrors.aliyun.com/ubuntu-ports/ xenial-security main*

*deb-src http://mirrors.aliyun.com/ubuntu-ports/ xenial-security main*

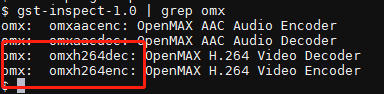
*deb http://mirrors.aliyun.com/ubuntu-ports/ xenial-security universe*

*deb-src http://mirrors.aliyun.com/ubuntu-ports/ xenial-security universe*

1. Execute command： sudo apt-get update
2. Execute command： sudo apt-get upgrade

#### Install the GStreamer component

1. .apt Mode installation, execute the command: （apt方式安装，执行命令：）
2. sudo apt-get install gstreamer1.0-alsa gstreamer1.0-clutter-3.0 gstreamer1.0-doc gstreamer1.0-espeak gstreamer1.0-fluendo-mp3 gstreamer1.0-libav gstreamer1.0-libav-dbg gstreamer1.0-nice gstreamer1.0-packagekit gstreamer1.0-plugins-bad gstreamer1.0-plugins-bad-dbg gstreamer1.0-plugins-bad-doc gstreamer1.0-plugins-base gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-base-dbg gstreamer1.0-plugins-base-doc gstreamer1.0-plugins-good gstreamer1.0-plugins-good-dbg gstreamer1.0-plugins-good-doc gstreamer1.0-plugins-ugly gstreamer1.0-plugins-ugly-dbg gstreamer1.0-plugins-ugly-doc gstreamer1.0-pocketsphinx gstreamer1.0-pulseaudio gstreamer1.0-tools gstreamer1.0-x gstreamer-qapt -y
3. sudo apt-get install libgstreamer-plugins-\* -y
4. sudo apt-get install libgstreamer1.0\* -y
5. Verify success of OMX, execute command：gst-inspect-1.0 | grep omx



#### Install OpenGL components

Execute the command as follows：

sudo apt-get install build-essential libgl1-mesa-dev -y

sudo apt-get install freeglut3-dev -y

sudo apt-get install libglew-dev libsdl2-dev libsdl2-image-dev libglm-dev libfreetype6-dev -y

#### **Compiling opencv 3.3.1 environment**

1. **Unzip source package: default to $home directory（解压源码包：默认在$HOME目录下）**

**tar -zxvf opencv-3.3.1.tgz**

1. **Install dependent environment：**

sudo apt-get install -y \

cmake \

libavcodec-dev \

libavformat-dev \

libavutil-dev \

libeigen3-dev \

libglew-dev \

libgtk2.0-dev \

libgtk-3-dev \

libjasper-dev \

libjpeg-dev \

libpng12-dev \

libpostproc-dev \

libswscale-dev \

libtbb-dev \

libtiff5-dev \

libv4l-dev \

libxvidcore-dev \

libx264-dev \

libx265-dev \

zlib1g-dev \

pkg-config

1. **Create installation path and compilation path:（创建安装路径和编译路径：）**

cd opencv-3.3.1

mkdir -p build/

cd build

sudo mkdir -p /usr/local/opencv

sudo chmod a+wrx -R /usr/local/opencv/

1. **Execute the compile command:（执行编译命令：）**
2. time cmake -D CMAKE\_BUILD\_TYPE=RELEASE \

-D CMAKE\_INSTALL\_PREFIX=/usr/local/opencv \

-D ENABLE\_FAST\_MATH=ON \

-D WITH\_CUBLAS=ON \

-D WITH\_LIBV4L=ON \

-D WITH\_GSTREAMER=ON \

-D WITH\_GSTREAMER\_0\_10=OFF \

-D WITH\_OPENGL=ON \

../

1. make install

#### Operating environment configuration of the above components（上述组件运行环境配置）

1. The environment variable in / etc / profile is as follows:）在/etc/profile中的环境变量，具体内容如下）

Open file

sudo vim /etc/profile

1. Add opencv config:

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/opencv/lib

export PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/usr/local/opencv/lib/pkgconfig

export PATH=$PATH:/usr/local/opencv/bin

1. Enable environment:（启用环境：）

source /etc/profile

#### **MultDecoder operation environment and execution environment**

1. % running process must be switched to root user on aikit local terminal for running and compiling(**源代码环境描述，目录结构如下：)**

└── MultDecoder

├── MultDecoderApp.cpp main program (主程序)

├── out cmake Compile dirctionary (cmake编译目录)

├── config ini profile management module(配置文件处理模块)

│   ├── include

│   ├── iniparser

│   ├── out

│   └── src

├── res Resources and profiles（资源及配置文件）

│ ├──  camera\_conf.h

├── lib Dependency Library

├── include Header file of dependent Library

├── external Open source library resources

1. **Project compilation steps(工程编译步骤：)**
2. **Configure compilation environment and execute(配置编译环境，执行:)**

source /etc/profile

1. **Compile source code：**
2. cd MultDecoder
3. Mkdir out
4. Cmake ..
5. make
6. **Running environment settings(运行环境设置：)**
7. **After compiling, verify the output to run app(编译完毕后，验证输出运行bin:)**

cd MultDecoder/bin

The related running bin file is as follows:(相关的运行bin文件如下

├── mult\_decoder\_app MultDecoder bin

1. **Start the mult\_decoder\_app**

**./mult\_decoder\_app**

Thank you for reading. I hope it will help you!(感谢阅览，希望对您有所帮助！)