

Python 调用 AXon OpenNI 使用说明

一、安装 Linux 平台的 AXonOpenNI SDK

以 Ubuntu16.04 为例

1.1 安装包名称格式

安装包名称: AXonOpenNI-Linux-x64-2.3.tar.bz2

1.2 安装过程

1. 安装所需驱动

```
sudo apt-get install libudev-dev libusb-1.0-0-dev
```

2. 安装图形化窗口

```
sudo apt-get install build-essential freeglut3 freeglut3-dev
```

3. 安装包解压

```
tar -jxvf AXonOpenNI-Linux-x64-2.3.tar.bz2
```

4. 执行安装脚本

```
sudo chmod +x install.sh
```

```
sudo ./install.sh
```

5. 设置环境变量

```
source OpenNIDevEnvironment
```

6. 编译 sample 文件(例如 AXon)

```
cd Samples/AXon
```

```
make
```

7. 运行可执行程序

```
cd Bin/x64-Release
```

```
sudo ./AXon
```

二、安装 python 库

1、opencv-python

```
pip install opencv-python
```

2、numpy

```
pip install numpy
```

3、安装 axonopenni 库（基于官方的 openni 库修改）

把 axonopenni 文件解压到/usr/lib/python3.5 文件夹下

三、环境配置

方法一、找到 AXonOpenNI-Linux-x64-2.3.tar.bz2 解压的文件夹，找到 Redist 文件夹，将该路径（如/home/myroom/AXonOpenNI-Linux-x64-2.3/Redist）添加到 python 的环境变量中。

方法二、打开 axonopenni 文件夹中的 openni2.py 文件，在第 27 行添加一行代码：

```
default_dll_directories.append("/home/myroom/AXonOpenNI-Linux-x64-2.3/Redist")
```

四、运行 python Demo 脚本

```
python3 SimpleDemo.py
```

五、AXon SDK 常用接口介绍

5.1 打开设备，创建流，获取流数据信息

```
openni2.initialize()
```

```

dev = openni2.Device.open_any()
dvs = dev.create_depth_stream()
cvs = dev.create_color_stream()
rvs = dev.create_ir_stream()
streams = [dvs, cvs, rvs]
for stream in streams:
    stream.start()
while True:
    openni2.wait_for_any_stream(streams)
    dframe = streams[0].read_frame()
    cframe = streams[1].read_frame()
    rframe = streams[2].read_frame()
for stream in streams:
    stream.stop()
    stream.close()
dev.close()
openni2.unload()

```

5.2 VideoMode 说明

VideoMode 表示数据流的格式，包括图像宽度、高度、像素格式和帧率。可以通过 `dev.get_sensor_info(sensor_type)` 来获得设备某一种 sensor(Color/IR/Depth) 支持的所有的 VideoMode，如：

```

info = dev.get_sensor_info(openni2.SENSOR_DEPTH)
print (info.sensorType, info.videoModes)

```

5.3 切换 VideoMode 模式

```

rgb_modes = cvs.get_sensor_info().videoModes
if len(rgb_modes) > 1:
    cvs.set_video_mode(rgb_modes[1])
vm = cvs.get_video_mode()

```

```
print("切换后的 RGB VideoMode:", vm)
```

5.4 获取设备信息

```
dev_info = dev.get_device_info()
```

5.5 获取设备固件版本号

```
from axonopenni import _openni2 as c_api
```

```
version = dev.get_property(c_api.ONI_DEVICE_PROPERTY_FIRMWARE_VERSION,  
                           c_api.AXonLinkFWVersion)
```

5.6 获取相机内外参数

```
camera=dev.get_property(c_api.AXONLINK_DEVICE_PROPERTY_GET_CAMERA_PARAM  
ETERS, c_api.AXonLinkCamParam)
```

5.7 获取设备序列号

```
serial = dev.get_property(c_api.ONI_DEVICE_PROPERTY_SERIAL_NUMBER,  
                          c_api.AXonLinkSerialNumber)
```

5.8 设置 rgb 和 depth 同步

```
dev.set_depth_color_sync_enabled(True)
```

5.9 设置 rgb 和 depth 对齐

设置对齐会增加主机端的 CPU 资源消耗。目前支持的同步模式见下表。注意，部分模式要求先[设置 RGB 和 Depth 同步](#)、或者 RGB 和 Depth 分辨率必须相同。

模式	说明	是否要求同步	RGB 和 Depth 分辨率是否必须相同
IMAGE_REGISTRATION_OFF	不对齐, Color 无去畸变校正	否	否
IMAGE_REGISTRATION_DEPTH_TO_COLOR	Depth 对齐到 Color, IR 不对齐, Color 去畸变校正	否	否
IMAGE_REGISTRATION_DEPTH_IR_TO_COLOR	Depth 和 IR 对齐到 Color, Color 去畸变校正	否	否
IMAGE_REGISTRATION_COLOR_TO_DEPTH	Color 对齐到 Depth 和 IR, Color 去畸变校正	是	是
IMAGE_REGISTRATION_COLOR_UNDISTORTION_ONLY	仅对 Color 去畸变校正	否	否

代码部分（具体可以查看 SimpleDemo 源码部分）：

```
mode=c_api.OniImageRegistrationMode.ONI_IMAGE_REGISTRATION_COLOR_TO_DEPTH
c_api.oniDeviceSetProperty(dev_handle,c_api.ONI_DEVICE_PROPERTY_IMAGE_REGISTRATION, ctypes.byref(mode), ctypes.sizeof(mode))
```

5.10 DSP 接口设置

DSP 接口设置一般不需要修改。

示例配置“去飞行像素”、“去飞行像素 2”、“3D 降噪”，可以根据需要设置。其他参数建议不要修改。

关于“去飞行像素”和“去飞行像素 2”的说明：

- “去飞行像素 2”的效果更好，但是会消耗主机 CPU 资源。默认是关闭的。

2. “去飞行像素”效果差一些，但是不消耗 CPU 资源。默认是打开的。

代码部分：

```
dsp = c_api.AXonDSPInterface()
dsp.FLYING = 1
dev.invoke(c_api.AXONLINK.AXONLINK_DEVICE_INVOKE_SET_DSP_DPINTERFACE,
           dsp)
```

5.11 设置左右镜像

```
cvs = dev.create_color_stream()
cvs.set_mirroring_enabled(True)
```

5.12 设置上下翻转

```
cvs.set_property(c_api.AXONLINK.AXONLINK_STREAM_PROPERTY_FLIP, 1)
```