3、 ROS+Opencv basic course

This lesson takes the Gemini2 camera as an example, the ordinary camera is similar.

3.1、Overview

Wiki：http://wiki.ros.org/cv_bridge/

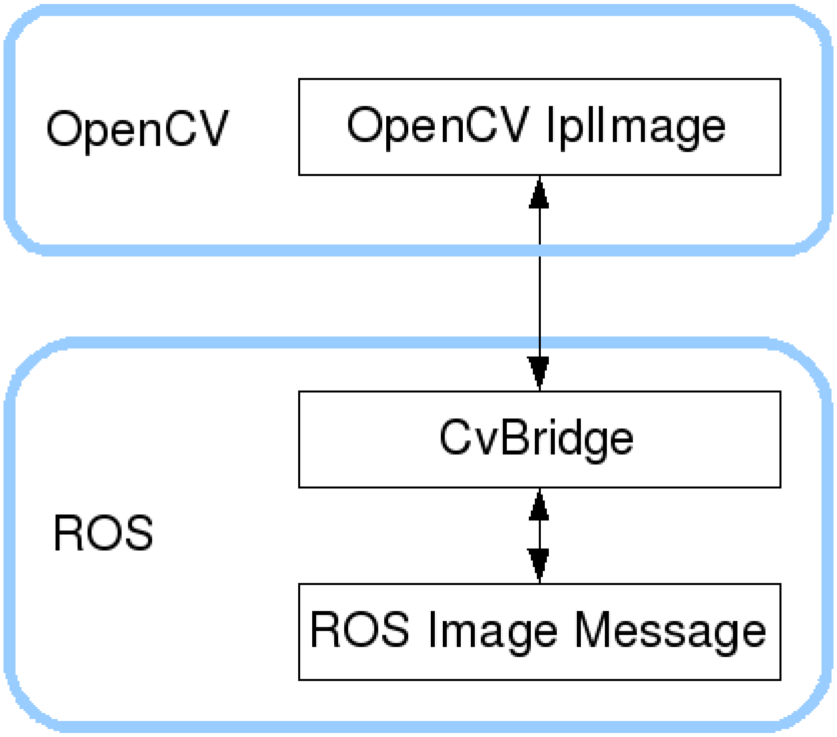
Tutorials：http://wiki.ros.org/cv_bridge/Tutorials

Source code：https://github.com/ros-perception/vision_opencv.git

Function package path：~/astra_ws/src/astra_visual

【CvBridge】is a ROS library, equivalent to a bridge between ROS and Opencv. It can perfectly convert and be converted image data format.

Opencv and ROS image data conversion is shown below：



This function package not only needs to use **[CvBridge**], but also needs **[Opencv]** and **[PCL]**, so we need to perform the following configuration.

- package.xml

Add following content.

```
    <build_depend>sensor_msgs</build_depend>
    <build_export_depend>sensor_msgs</build_export_depend>
    <exec_depend>sensor_msgs</exec_depend>

    <build_depend>std_msgs</build_depend>
    <build_export_depend>std_msgs</build_export_depend>
    <exec_depend>std_msgs</exec_depend>

    <build_depend>cv_bridge</build_depend>
    <build_export_depend>cv_bridge</build_export_depend>
    <exec_depend>cv_bridge</exec_depend>

    <exec_depend>image_transport</exec_depend>
```

【cv_bridge】： Image conversion dependent package.

【transbot_msgs】： Custom message dependency package.

- CMakeLists.txt

This file has a lot of configuration content, please check the source file for specific content.

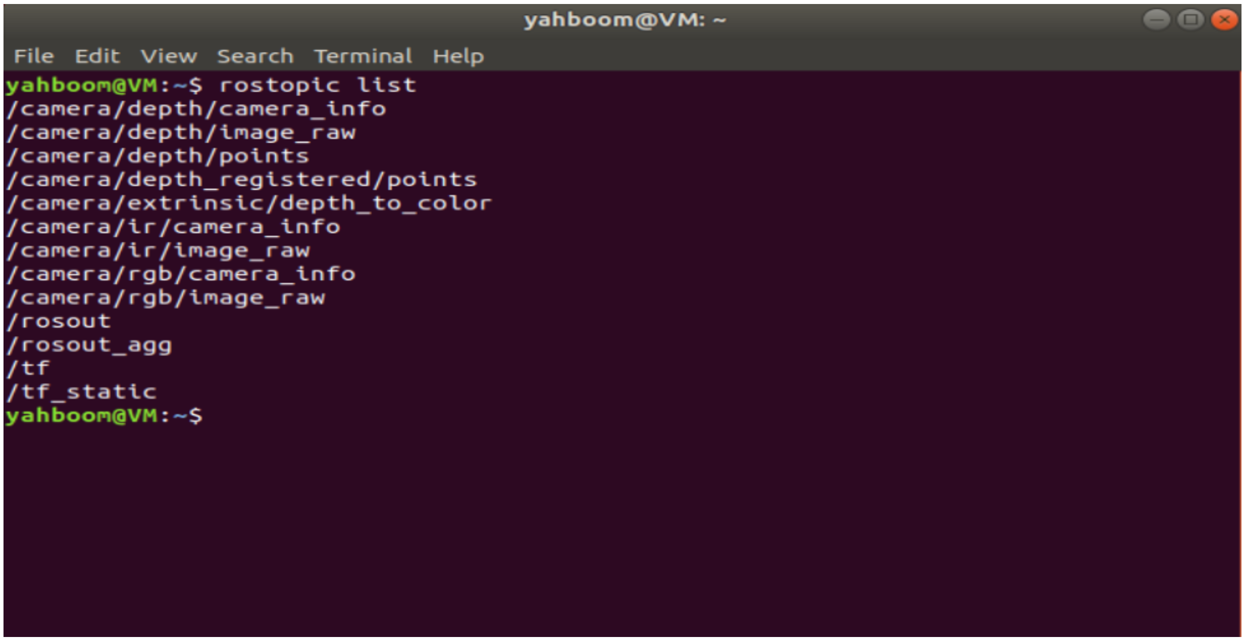
3.2、Gemini2

3.2.1、Start up Gemini2 camera

```
roslaunch orbbec_camera gemini2.launch
```

View topic

```
rostopic list
```



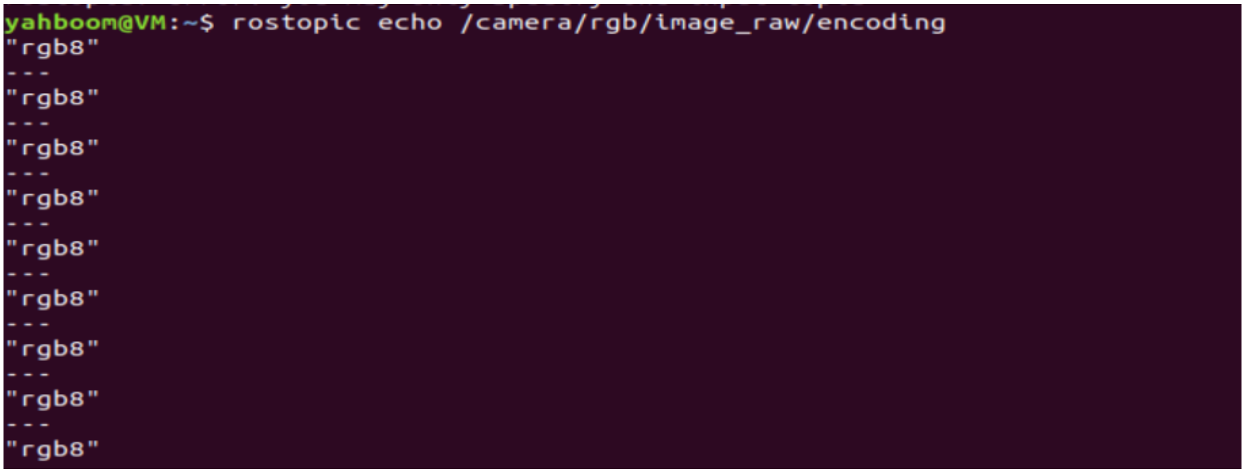
```
yahboom@VM:~$ rostopic list
/camera/depth/camera_info
/camera/depth/image_raw
/camera/depth/points
/camera/depth_registered/points
/camera/extrinsic/depth_to_color
/camera/ir/camera_info
/camera/ir/image_raw
/camera/rgb/camera_info
/camera/rgb/image_raw
/rosout
/rosout_agg
/tf
/tf_static
yahboom@VM:~$
```

Common topics are as follows

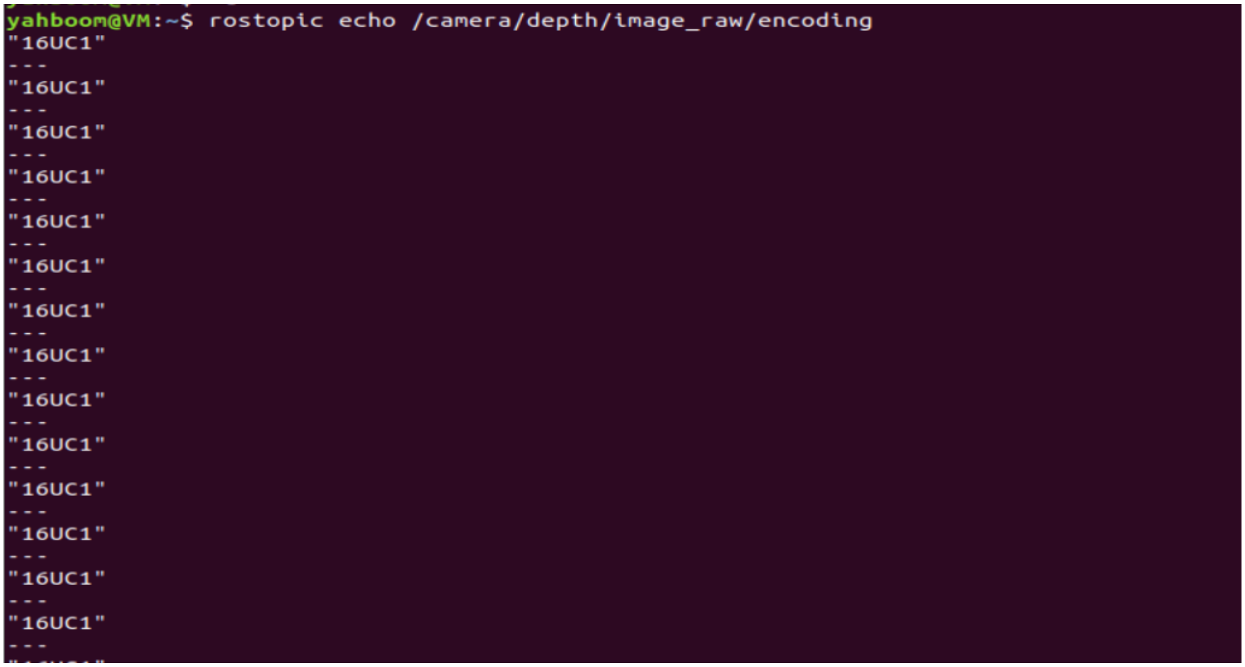
Topic name	Data type
/camera/depth/image_raw	sensor_msgs/Image
/camera/rgb/image_raw	sensor_msgs/Image
/camera/ir/image_raw	sensor_msgs/Image
/camera/depth/points	sensor_msgs/PointCloud2

View the encoding format of the topic: rostopic echo +【topic】+encoding, for example

```
rostopic echo /camera/rgb/image_raw/encoding
rostopic echo /camera/depth/image_raw/encoding
```



```
yahboom@VM:~$ rostopic echo /camera/rgb/image_raw/encoding
"rgb8"
---
"rgb8"
---
"rgb8"
---
"rgb8"
---
"rgb8"
---
"rgb8"
---
"rgb8"
---
"rgb8"
---
"rgb8"
```



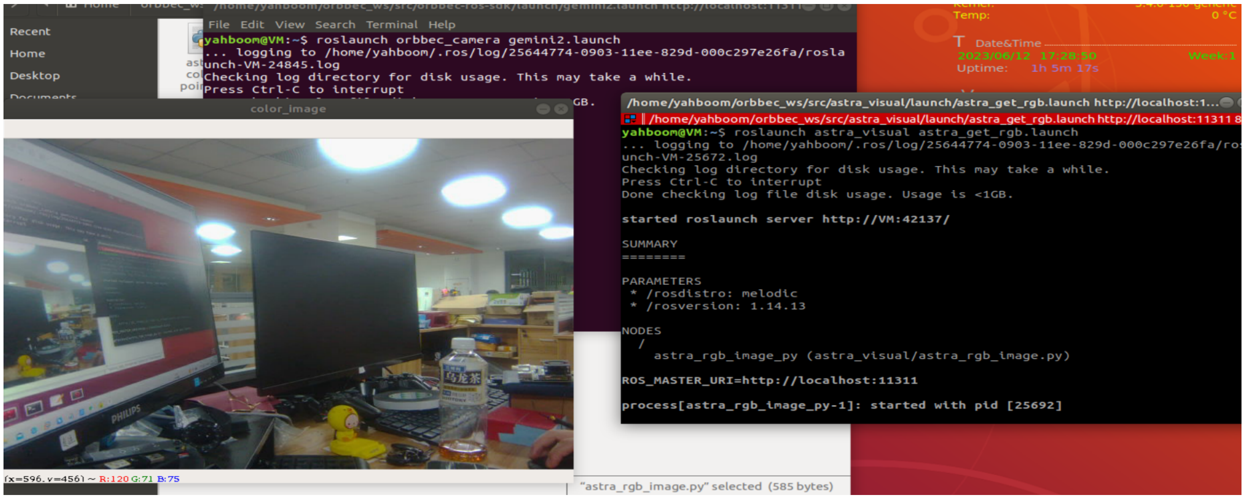
```
yahboom@VM:~$ rostopic echo /camera/depth/image_raw/encoding
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
"16UC1"
---
```

3.2.2、Start the color map subscription node

```
roslaunch orbbec_camera gemini2.launch          # start camera
roslaunch astra_visual astra_get_rgb.launch     #Start function
```

The node launched by the content of the launch file is one of the following two options,

```
astra_rgb_image.py          # py
astra_rgb_image             # C++
```



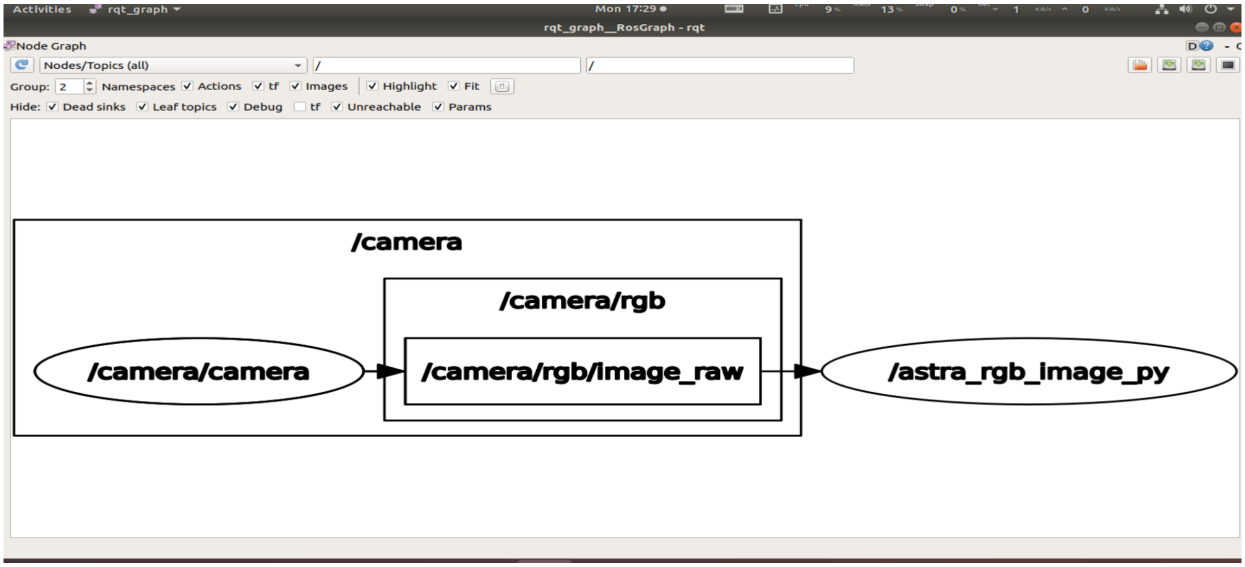
```
yahboom@VM:~$ roslaunch orbbec_camera gemini2.launch
... logging to /home/yahboom/.ros/log/25644774-0903-11ee-829d-000c297e26fa/rosla
unch-VM-24845.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
```

```
yahboom@VM:~$ roslaunch astra_visual astra_get_rgb.launch
... logging to /home/yahboom/.ros/log/25644774-0903-11ee-829d-000c297e26fa/ro
unch-VM-25672.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://VM:42137/

SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.13

NODES
  /
    astra_rgb_image_py (astra_visual/astra_rgb_image.py)

ROS_MASTER_URI=http://localhost:11311

process[astra_rgb_image_py-1]: started with pid [25692]
```

Recent
Home
Desktop
File Edit View Search Terminal Help
color_image
/home/yahboom/orbbec_ws/src/astra_visual/launch/astra_get_rgb.launch http://localhost:1...
/home/yahboom/orbbec_ws/src/astra_visual/launch/astra_get_rgb.launch http://localhost:11311
Temp: 0 °C
Date&Time
2023/06/12 17:28:50 Week:1
Uptime: 1h 5m 17s
"astra_rgb_image.py" selected (585 bytes)

View node graph

```
rqt_graph
```



Nodes/Topics (all) / /

Group: 2 ☑ Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit

Hide: ☑ Dead sinks ☑ Leaf topics ☑ Debug □ tf ☑ Unreachable ☑ Params

- py code analysis

Create subscribers: The subscribed topic is 【"/camera/rgb/image_raw"】, the data type is 【Image】, and the callback function is 【topic()】

```
sub = rospy.Subscriber("/camera/rgb/image_raw", Image, topic)
```

Use 【CvBridge】 for data conversion to ensure that the encoding format is correct.

```
frame = bridge.imgmsg_to_cv2(msg, "bgr8")
```

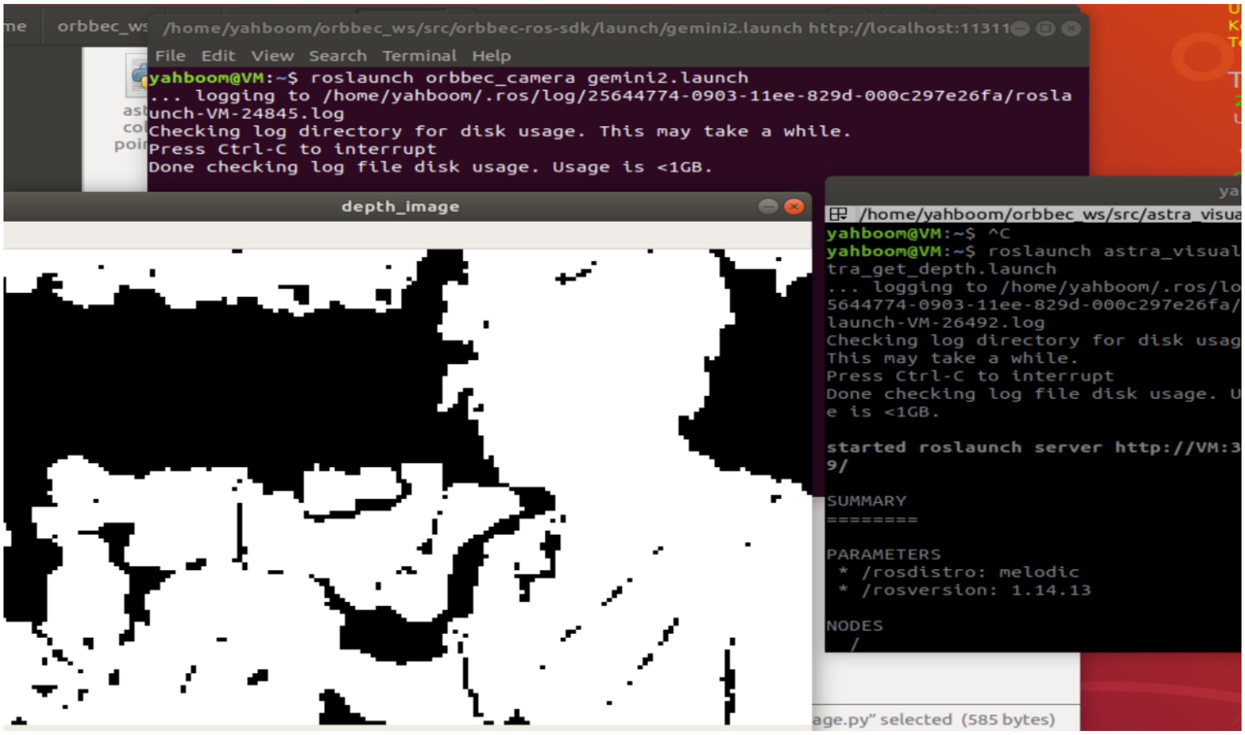
- c++ code analysis

Similar to py code

```
//Create a receiver.
ros::Subscriber subscriber = n.subscribe<sensor_msgs::Image>
("/camera/rgb/image_raw", 10, RGB_Callback);
// Create cv_bridge example
cv_bridge::CvImagePtr cv_ptr;
// Data conversion
cv_ptr = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::BGR8);
```

3.2.3、Start the depth map subscription node

```
roslaunch orbbec_camera gemini2.launch          #
roslaunch astra_visual astra_get_depth.launch    # launch
```



/home/yahboom/orbbec_ws/src/orbbec-ros-sdk/launch/gemini2.launch http://localhost:11311
File Edit View Search Terminal Help
depth_image
/home/yahboom/orbbec_ws/src/astra_visua
age.py" selected (585 bytes)

```
yahboom@VM:~$ roslaunch orbbec_camera gemini2.launch
... logging to /home/yahboom/.ros/log/25644774-0903-11ee-829d-000c297e26fa/rosla
unch-VM-24845.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
```

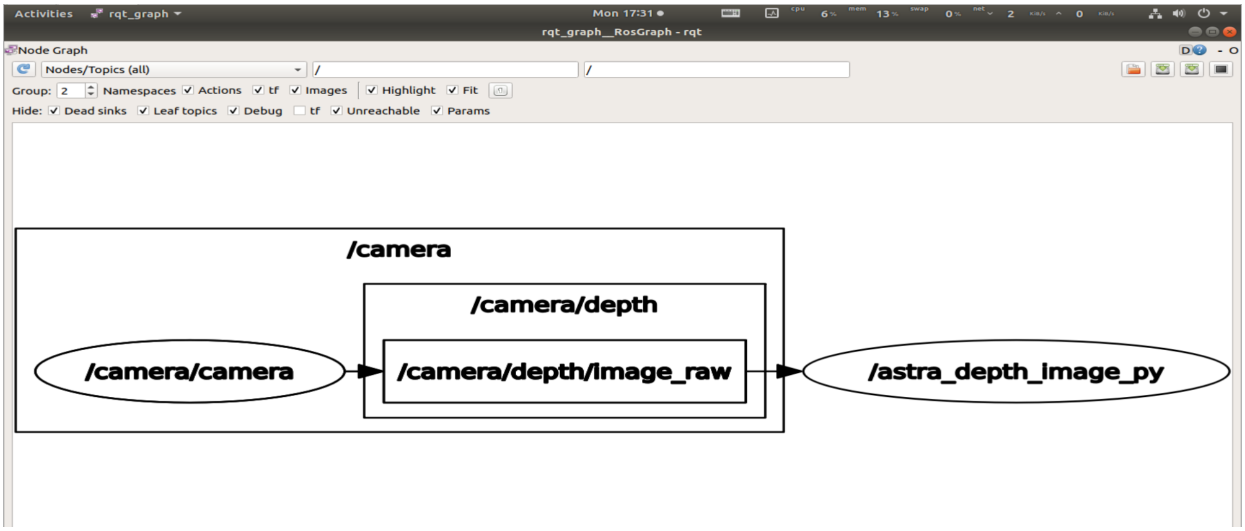
```
yahboom@VM:~$ ^C
yahboom@VM:~$ roslaunch astra_visual
tra_get_depth.launch
... logging to /home/yahboom/.ros/lo
5644774-0903-11ee-829d-000c297e26fa/
launch-VM-26492.log
Checking log directory for disk usag
This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. U
e is <1GB.

started roslaunch server http://VM:3
9/

SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.13

NODES
  /
```

The node launched by the content of the launch file is one of the following two options,

```
astra_depth_image.py        # py
astra_depth_image           # C++
```

View node graph

```
rqt_graph
```



Nodes/Topics (all)   /   /

Group: 2   Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit

Hide: ☑ Dead sinks ☑ Leaf topics ☑ Debug □ tf ☑ Unreachable ☑ Params

- py code analysis

Create subscribers: The subscribed topic is ["/camera/depth/image_raw"], the data type is [Image], and the callback function is [topic()]

```
sub = rospy.Subscriber("/camera/depth/image_raw", Image, topic)
```

Use 【CvBridge】 for data conversion to ensure that the encoding format is correct.。

```
# Encoding format
encoding = ['16UC1', '32FC1']
# Can switch different encoding formats to test the effect
frame = bridge.imgmsg_to_cv2(msg, encoding[1])
```

- c++ code analysis

Similar to py code

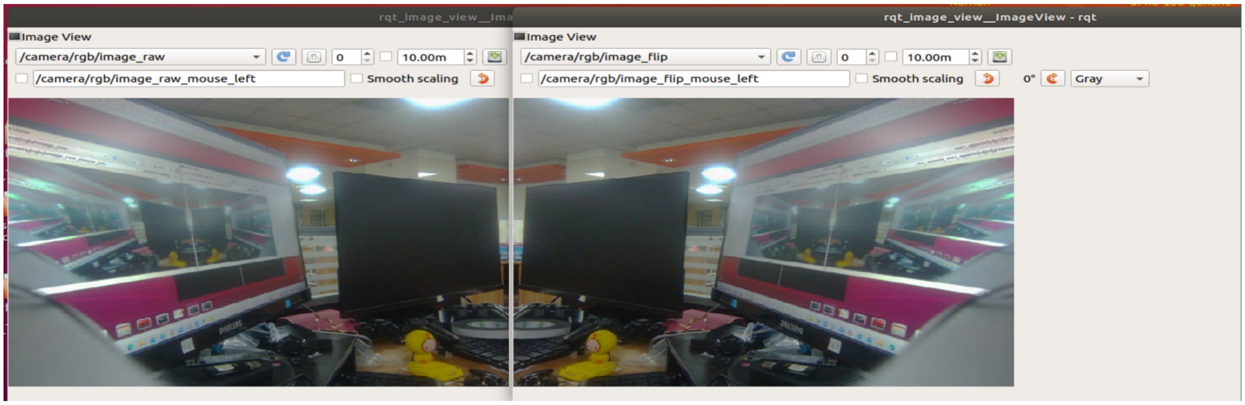
```
//Create a receiver.
ros::Subscriber subscriber = n.subscribe<sensor_msgs::Image>
("/camera/depth/image_raw", 10, depth_Callback);
// Create cv_bridge example
cv_bridge::CvImagePtr cv_ptr;
// Data conversion
cv_ptr = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::TYPE_16UC1);
```

3.2.4、Start color image inversion

```
roslaunch orbbec_camera gemini2.launch          #turn on camera
roslaunch astra_visual astra_image_flip.launch    #open function
```

image viewing

```
rqt_image_view      #open the first
rqt_image_view      #open the second
```


/camera/rgb/image_raw
/camera/rgb/image_raw_mouse_left
/camera/rgb/image_flip
/camera/rgb/image_flip_mouse_left

- py code analysis

Two subscribers and two publishers are created here, one for processing general image data and one for processing compressed image data.

1）Create subscriber

The subscribed topic is【"/camera/rgb/image_raw"】, the data type is【Image】, and the callback function is【topic()】。

2）Create publisher

The published topic is【"/camera/rgb/image_flip"】, the data type is【Image】, Queue size【10】。

```
sub_img = rospy.Subscriber("/camera/rgb/image_raw", Image, topic)
pub_img = rospy.Publisher("/camera/rgb/image_flip", Image, queue_size=10)
```

3) Callback function

```
# Normally image transmission processing
def topic(msg):
    if not isinstance(msg, Image):
        return
    bridge = CvBridge()
    frame = bridge.imgmsg_to_cv2(msg, "bgr8")
    # Opencv processing image
    frame = cv.resize(frame, (640, 480))
    frame = cv.flip(frame, 1)
    # opencv mat ->  ros msg
    msg = bridge.cv2_to_imgmsg(frame, "bgr8")
    # After image processing is complete, publish directly
    pub_img.publish(msg)
```