

1. Environment setup

This course takes **Ubuntu20.04+ros-foxy** as an example to explain how to set up an environment to use the camera in a ros2 environment.

1. Install related dependencies

Terminal input,

```
sudo apt install libgflags-dev nlohmann-json3-dev libgoogle-glog-dev ros-foxy-  
image-transport ros-foxy-image-publisher
```

Here foxy** is modified according to the actual ros2 version**. If it is galactic, just change it to galactic.

2. Compile function package

1) Create a workspace

Take the workspace name orbbec_ws created in the ~ directory as an example.

```
mkdir orbbec_ws  
cd orbbec_ws  
mkdir src
```

2) Copy the function package to the workspace

Unzip the file, copy and paste the folder (function package) under src into the ~/orbbec_ws/src directory you just created.

3), compile

Terminal input,

```
cd ~/orbbec_ws  
colcon build
```

4), add environment variables

Terminal input,

```
echo "source ~/orbbec_ws/install/setup.bash" >> ~/.bashrc
```

3. Install udev rules

Terminal input,

```
cd ~/orbbec_ws/src/OrbbecSDK_ROS2/orbbec_camera/scripts  
sudo bash install.sh
```

Enter the following command to check whether the rule file is successfully loaded and the camera is bound.

```
#astraproplus
ll /dev/astro_pro_plus
#geminii2
ll /dev/OrbbecGeminii2
```

```
yahboom@VM:~/Desktop$ ll /dev/astro_pro_plus
lrwxrwxrwx 1 root root 15 11月  6 15:59 /dev/astro_pro_plus -> bus/usb/003/011
yahboom@VM:~/Desktop$
```

```
yahboom@VM:~/Desktop$ ll /dev/OrbbecGeminii2
lrwxrwxrwx 1 root root 6 11月 10 14:55 /dev/OrbbecGeminii2 -> video5
```

If the above picture appears, it means success.

4. Run the camera and view the images

Terminal input,

```
#astraproplus camera launch
ros2 launch orbbec_camera astra.launch.xml
#geminii2 camera launch
ros2 launch orbbec_camera geminii2.launch.py
```

Enter the following command to view topic information,

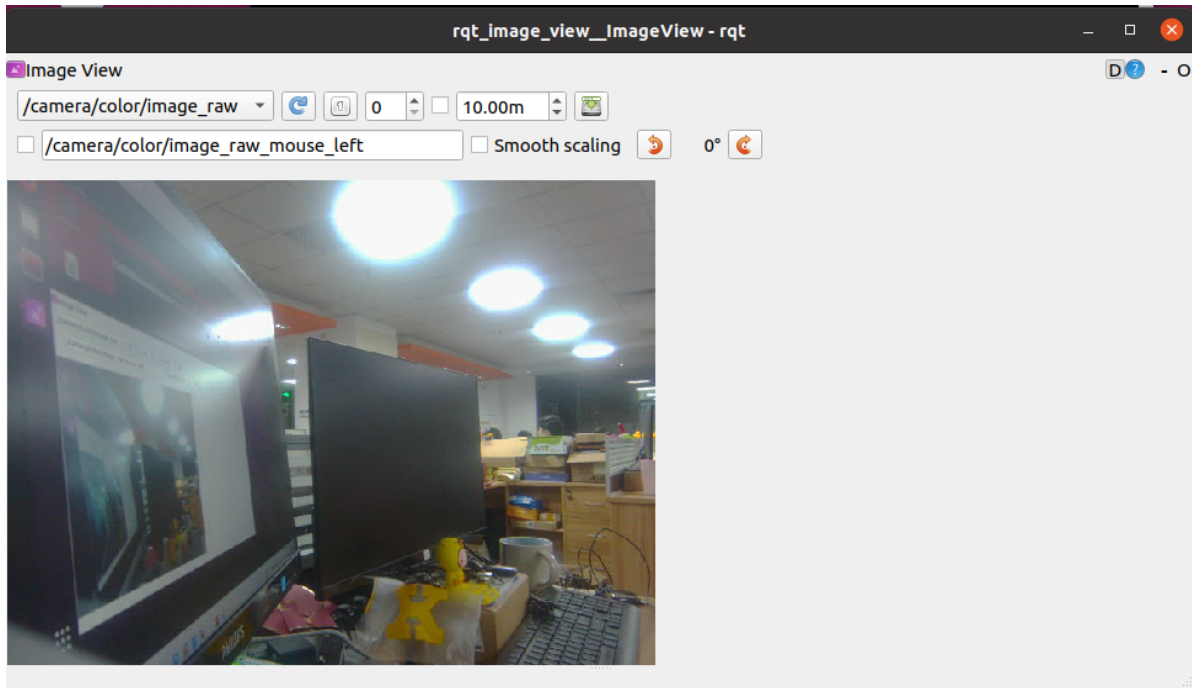
```
ros2 topic list
```

```
yahboom@VM:~/Desktop$ ros2 topic list
/camera/color/camera_info
/camera/color/image_raw
/camera/color/image_raw/compressed
/camera/color/image_raw/compressedDepth
/camera/color/image_raw/theora
/camera/depth/camera_info
/camera/depth/image_raw
/camera/depth/image_raw/compressed
/camera/depth/image_raw/compressedDepth
/camera/depth/image_raw/theora
/camera/depth/points
/camera/depth_registered/points
/camera/ir/camera_info
/camera/ir/image_raw
/camera/ir/image_raw/compressed
/camera/ir/image_raw/compressedDepth
/camera/ir/image_raw/theora
/parameter_events
/rosout
/tf
/tf_static
```

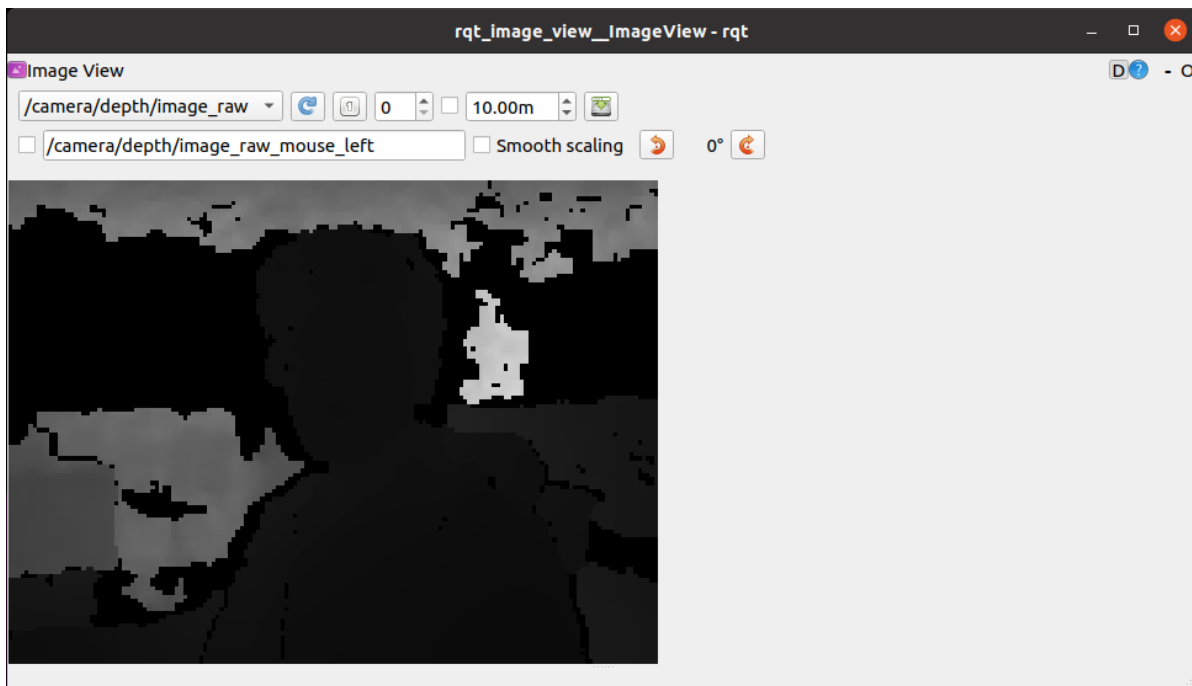
Use the rqt_image_view tool to view images, terminal input,

```
ros2 run rqt_image_view rqt_image_view
```

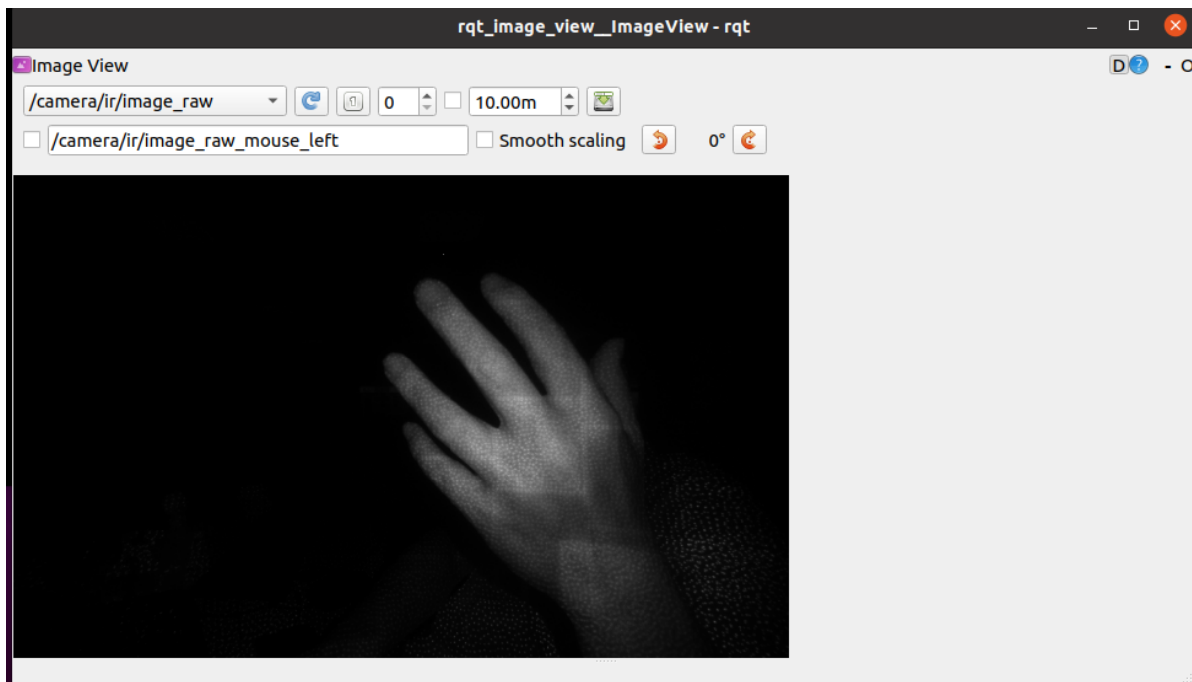
color picture



Depth map



Infrared IR image



Just select the topic in the upper left corner.

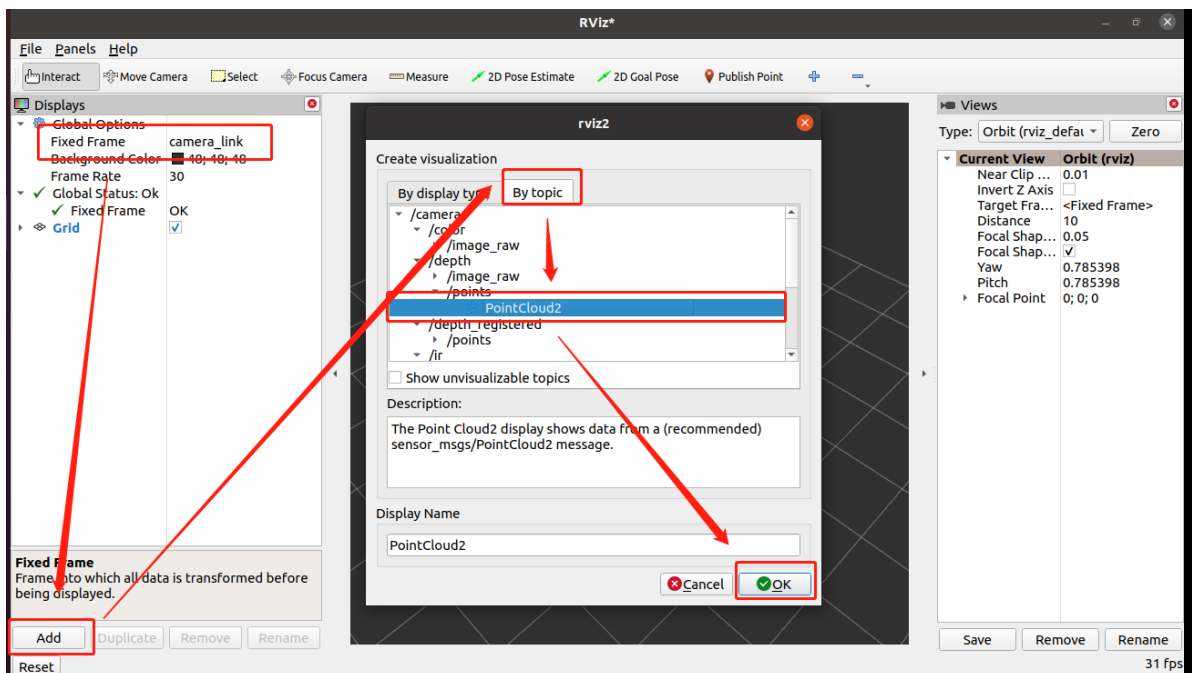
5. Run the camera and view the point cloud image

Terminal input,

```
ros2 launch orbbec_camera gemini2.launch.py
```

You can see the point cloud data released by the camera in rviz and enter it in the terminal.

```
rviz2
```



After starting rviz, set up the visual point cloud data as shown in the figure above.

