

1.Environmental construction

This course takes Ubuntu 20.04+ros foxy as an example to illustrate how to build an environment using the camera in the ROS2 environment.

2.Install related dependency libraries

Input following command:

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

Here, foxy is modified based on the actual version of ROS2 . If it is galactic, it can be replaced with galactic.

3.Compile Feature Pack

1) Create a workspace

To create a workspace named orbbec in the~directory_ Taking ws as an example,

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

2)Copy feature packs to workspace

Unzip the file, copy and paste the folder (feature pack) under src into the~/orbbec you just created_ Under the ws/src directory.

3)Compile

Input following command:

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

4)Add environment variables

Input following command:

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

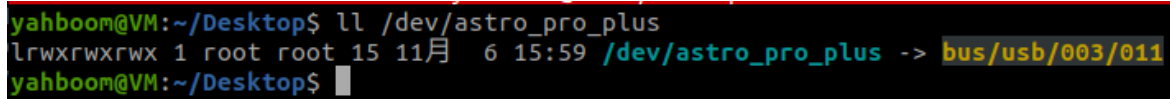
4.Install udev rules

Input following command:

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

Enter the following command to check if the rule file was successfully loaded and the camera was bound.

```
ll /dev/astro_pro_plus
```



```
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$
```

If the above image appears, it indicates success.

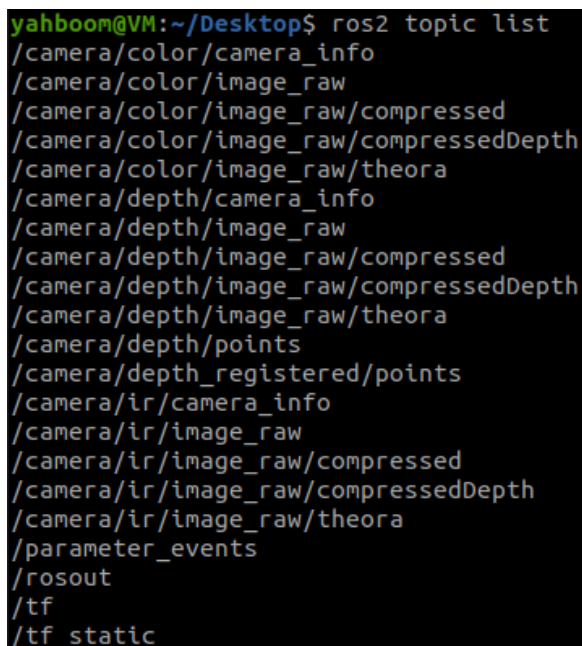
5.Run the camera and view the image

Input following command:

```
#Start up astraproplus
ros2 launch orbbec_camera astra.launch.xml
#Start up gemini2
ros2 launch orbbec_camera gemini2.launch.xml
```

Input following command to view topic information:

```
ros2 topic list
```

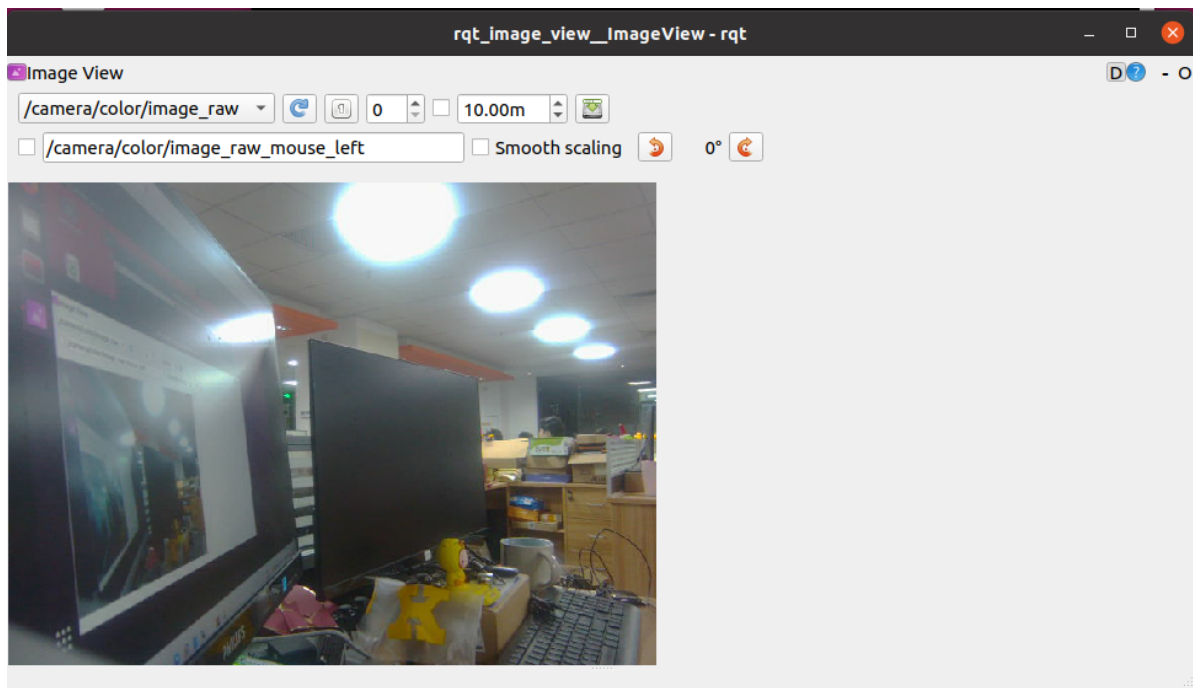


```
yahboom@VM:~/Desktop$ ros2 topic list
/camera/color/camera_info
/camera/color/image_raw
/camera/color/image_raw/compressed
/camera/color/image_raw/compressedDepth
/camera/color/image_raw/theora
/camera/depth/camera_info
/camera/depth/image_raw
/camera/depth/image_raw/compressed
/camera/depth/image_raw/compressedDepth
/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
```

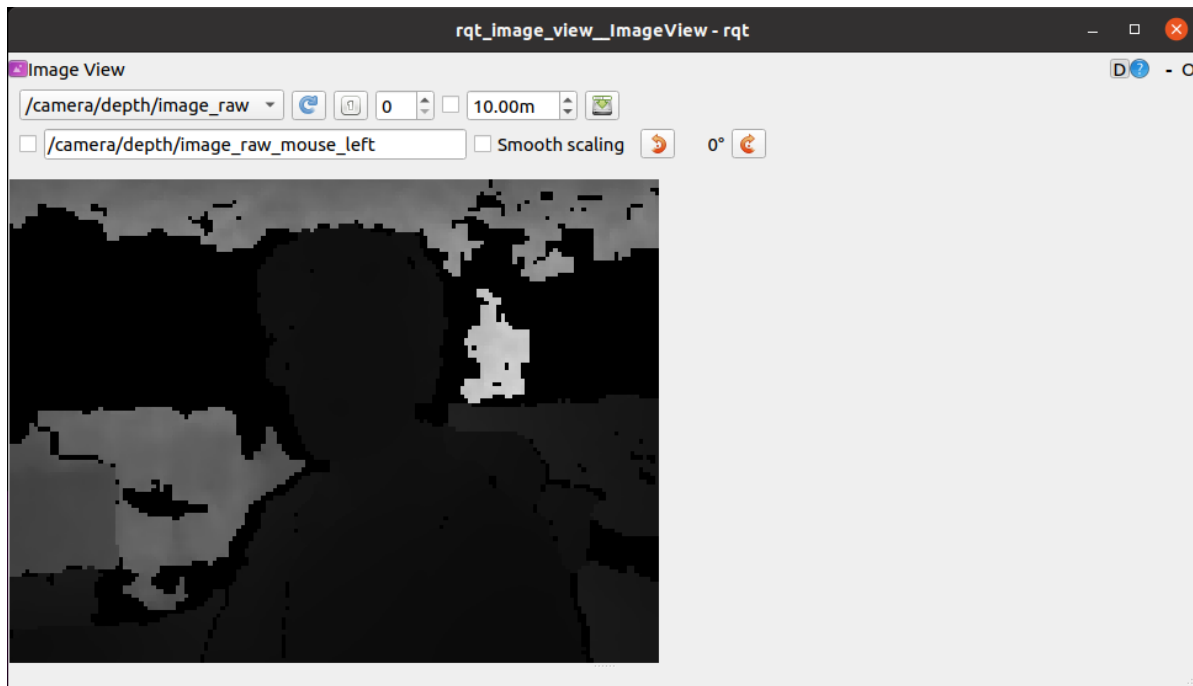
Using rqt_Image_View tool to view images. Input following command:

```
ros2 run rqt_image_view rqt_image_view
```

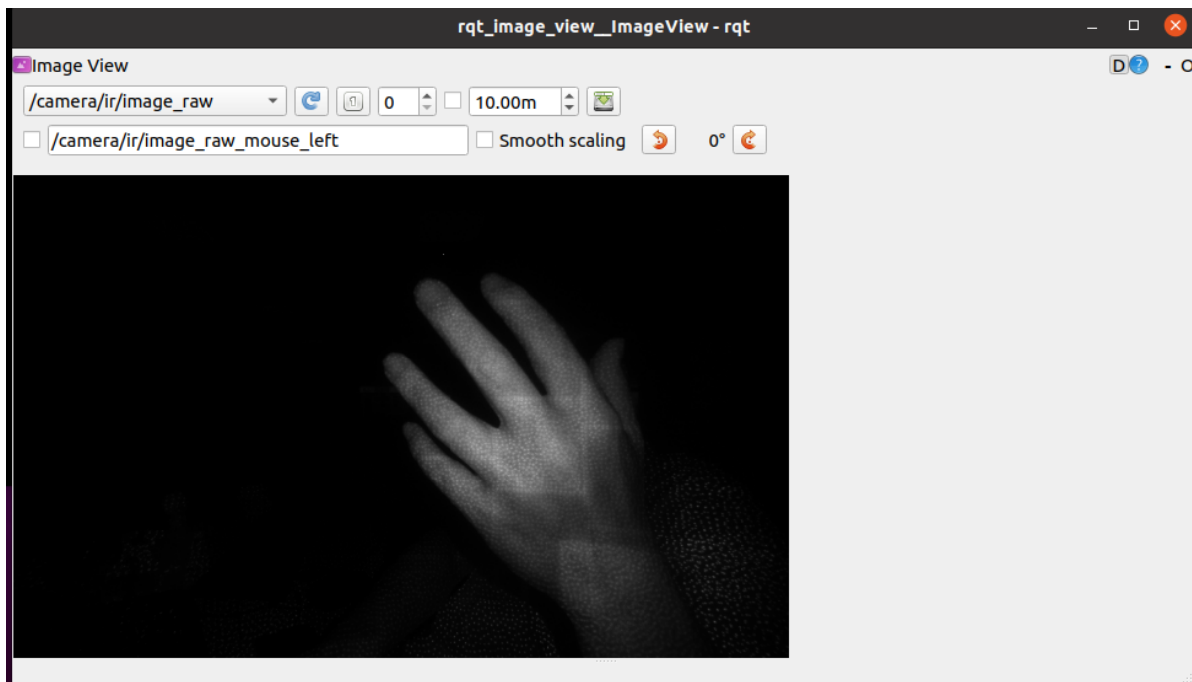
Color Map



Depth map



Infrared IR image



Select the topic in the upper left corner.

6. Run the camera and view the point cloud image

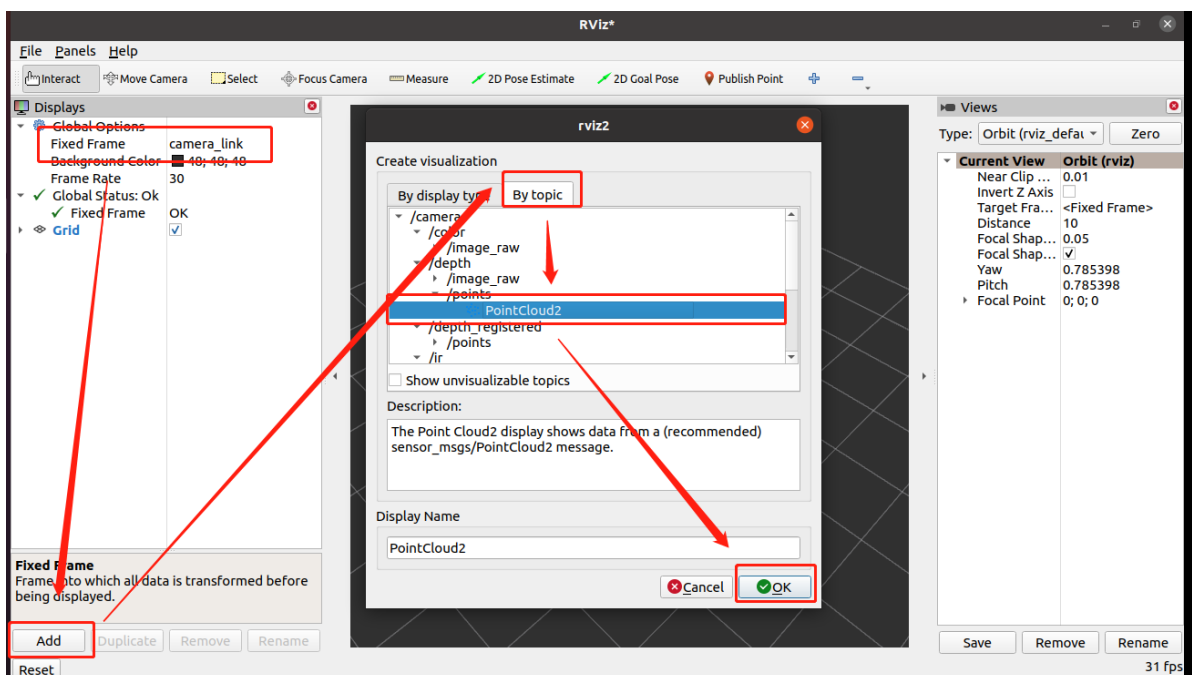
Input following command:

```
ros2 launch orbbec_camera dabai_dcw2.launch.py
```

You can see the point cloud data published by the camera in rviz.

Input following command:

```
rviz2
```



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

