1.Depth camera use

Before driving the depth camera, the astra camera device needs to be recognized on the host machine; when entering the docker container, the astra device needs to be mounted to recognize the camera in the docker container. The supporting host has already set up an environment and does not require additional configuration. If it is on a new host, you need to add a rule file. The adding method is very simple. Copy the /etc/udev.rules.d/56-orbbec-usb.rules file under the host to the /etc/udev.rules.d directory in the new environment, and then restart it.

1. Program function description

After the program is run, drive the Astra camera to obtain color RGB, depth, infrared IR image information and depth point cloud information.

2. Program code reference path

After entering the docker container, the location of the source code of this function is:

```
/root/yahboomcar_ros2_ws/software/library_ws/src/ros2_astra_camera/astra_camera/launch/
```

3. Program startup

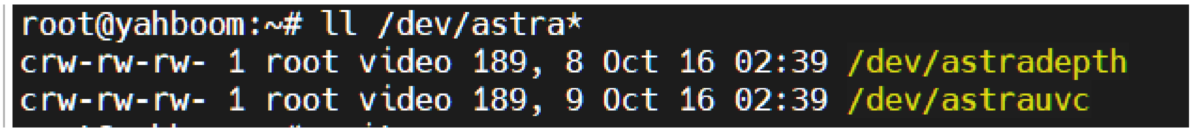
launch startup command,

launch file	Camera model
ros2 launch astra_camera astra_pro.launch.xml	Astrapro
ros2 launch astra_camera astro_pro_plus.launch.xml	Astraproplus
ros2 launch astra_camera astra.launch.xml	Astramini

Here we take starting the Astraproplus camera as an example. After entering the docker container, enter in the terminal:

```
ll /dev/astra*
```

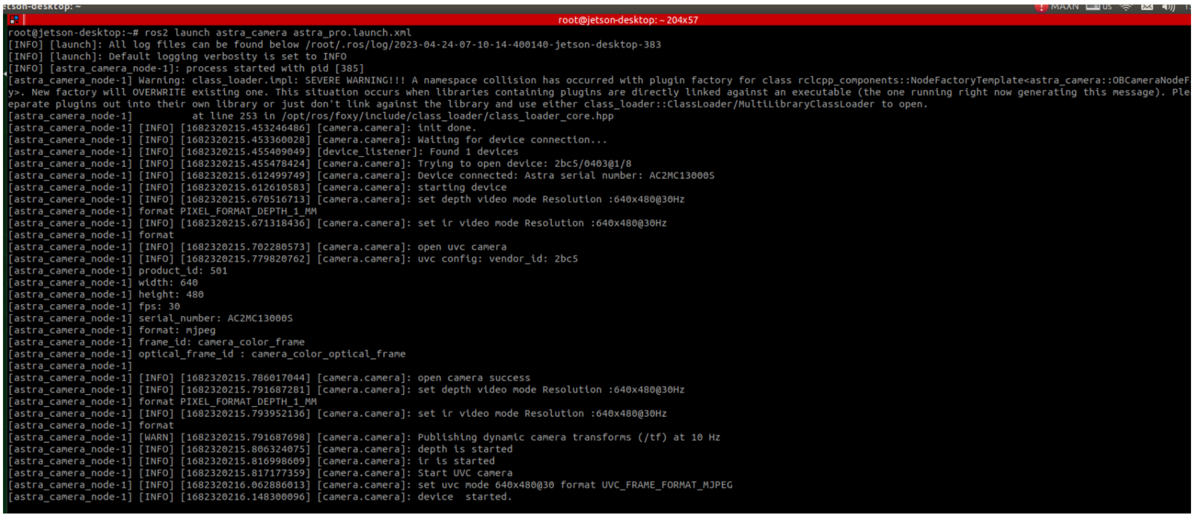
This step is to check whether the Astra camera device can be recognized in Docker. If the following content appears, it means that the Astraproplus camera is successfully mounted into the Docker container:



```
root@yahboom:~# ll /dev/astra*
crw-rw-rw- 1 root video 189, 8 Oct 16 02:39 /dev/astradepth
crw-rw-rw- 1 root video 189, 9 Oct 16 02:39 /dev/astrauvc
```

Enter in the Docker terminal,

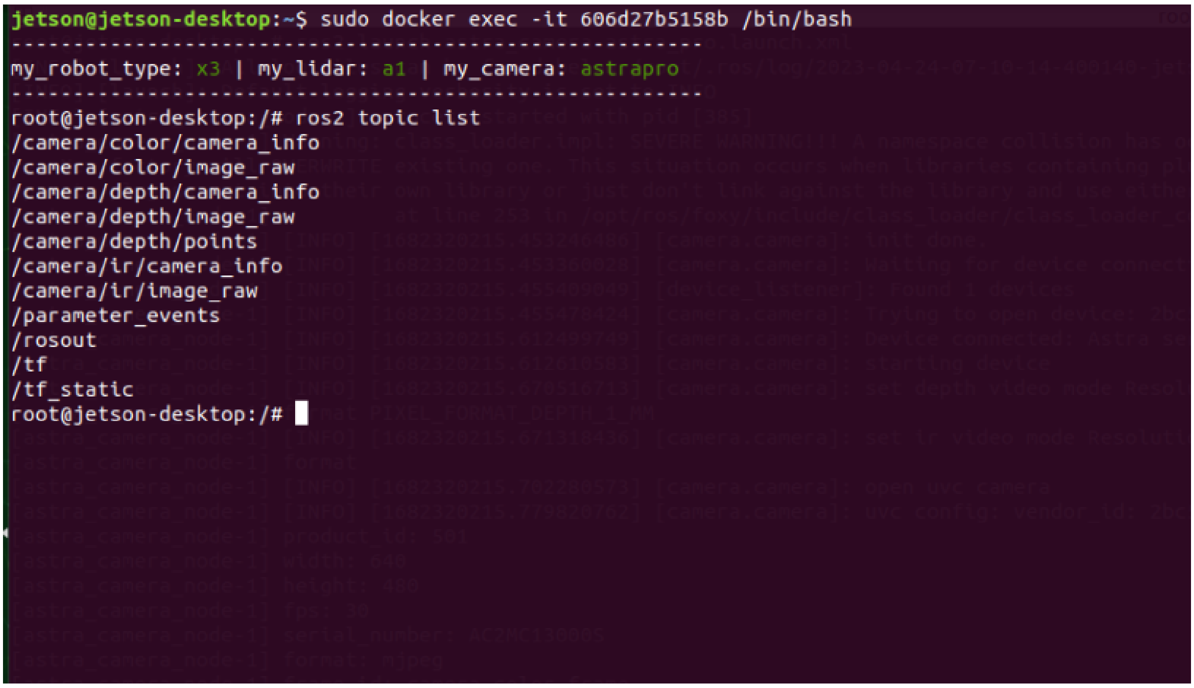
```
ros2 launch astra_camera astro_pro_plus.launch.xml
```



```
root@jetson-desktop:~# ros2 launch astra_camera astra_pro.launch.xml
[INFO] [launch]: All log files can be found below /root/.ros/log/2023-04-24-07-10-14-400140-jetson-desktop-383
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [astra_camera_node-1]: process started with pid [385]
[astra_camera_node-1] Warning: class_loader.impl: SEVERE WARNING!!! A namespace collision has occurred with plugin factory for class rclcpp_components::NodeFactoryTemplate<astra_camera::OBCameraNodeF
y>. New factory will OVERWRITE existing one. This situation occurs when libraries containing plugins are directly linked against an executable (the one running right now generating this message). Ple
eparate plugins out into their own library or just don't link against the library and use either class_loader::ClassLoader/MultiLibraryClassLoader to open.
[astra_camera_node-1]          at line 253 in /opt/ros/foxy/include/class_loader/class_loader_core.hpp
[astra_camera_node-1] [INFO] [1682320215.453246486] [camera.camera]: init done.
[astra_camera_node-1] [INFO] [1682320215.453360028] [camera.camera]: Waiting for device connection...
[astra_camera_node-1] [INFO] [1682320215.455409049] [device_listener]: Found 1 devices
[astra_camera_node-1] [INFO] [1682320215.455478424] [camera.camera]: Trying to open device: 2bc5/0403@1/8
[astra_camera_node-1] [INFO] [1682320215.612499749] [camera.camera]: Device connected: Astra serial number: AC2MC13000S
[astra_camera_node-1] [INFO] [1682320215.612610583] [camera.camera]: starting device
[astra_camera_node-1] [INFO] [1682320215.670516713] [camera.camera]: set depth video mode Resolution :640x480@30Hz
[astra_camera_node-1] format PIXEL_FORMAT_DEPTH_1_MM
[astra_camera_node-1] [INFO] [1682320215.671318436] [camera.camera]: set ir video mode Resolution :640x480@30Hz
[astra_camera_node-1] format
[astra_camera_node-1] [INFO] [1682320215.702280573] [camera.camera]: open uvc camera
[astra_camera_node-1] [INFO] [1682320215.779820762] [camera.camera]: uvc config: vendor_id: 2bc5
[astra_camera_node-1] product_id: 501
[astra_camera_node-1] width: 640
[astra_camera_node-1] height: 480
[astra_camera_node-1] fps: 30
[astra_camera_node-1] serial_number: AC2MC13000S
[astra_camera_node-1] format: mjpeg
[astra_camera_node-1] frame_id: camera_color_frame
[astra_camera_node-1] optical_frame_id : camera_color_optical_frame
[astra_camera_node-1]
[astra_camera_node-1] [INFO] [1682320215.786017044] [camera.camera]: open camera success
[astra_camera_node-1] [INFO] [1682320215.791687281] [camera.camera]: set depth video mode Resolution :640x480@30Hz
[astra_camera_node-1] format PIXEL_FORMAT_DEPTH_1_MM
[astra_camera_node-1] [INFO] [1682320215.793952136] [camera.camera]: set ir video mode Resolution :640x480@30Hz
[astra_camera_node-1] format
[astra_camera_node-1] [WARN] [1682320215.791687698] [camera.camera]: Publishing dynamic camera transforms (/tf) at 10 Hz
[astra_camera_node-1] [INFO] [1682320215.806324075] [camera.camera]: depth is started
[astra_camera_node-1] [INFO] [1682320215.816998609] [camera.camera]: ir is started
[astra_camera_node-1] [INFO] [1682320215.817177359] [camera.camera]: Start UVC camera
[astra_camera_node-1] [INFO] [1682320216.062886013] [camera.camera]: set uvc mode 640x480@30 format UVC_FRAME_FORMAT_MJPEG
[astra_camera_node-1] [INFO] [1682320216.148300096] [camera.camera]: device  started.
```

You can use the following command to view the topic, enter in the Docker terminal,

```
ros2 topic list
```



```
jetson@jetson-desktop:~$ sudo docker exec -it 606d27b5158b /bin/bash
-------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
-------------------------------------------------------
root@jetson-desktop:/# ros2 topic list
/camera/color/camera_info
/camera/color/image_raw
/camera/depth/camera_info
/camera/depth/image_raw
/camera/depth/points
/camera/ir/camera_info
/camera/ir/image_raw
/parameter_events
/rosout
/tf
/tf_static
root@jetson-desktop:/# 
```

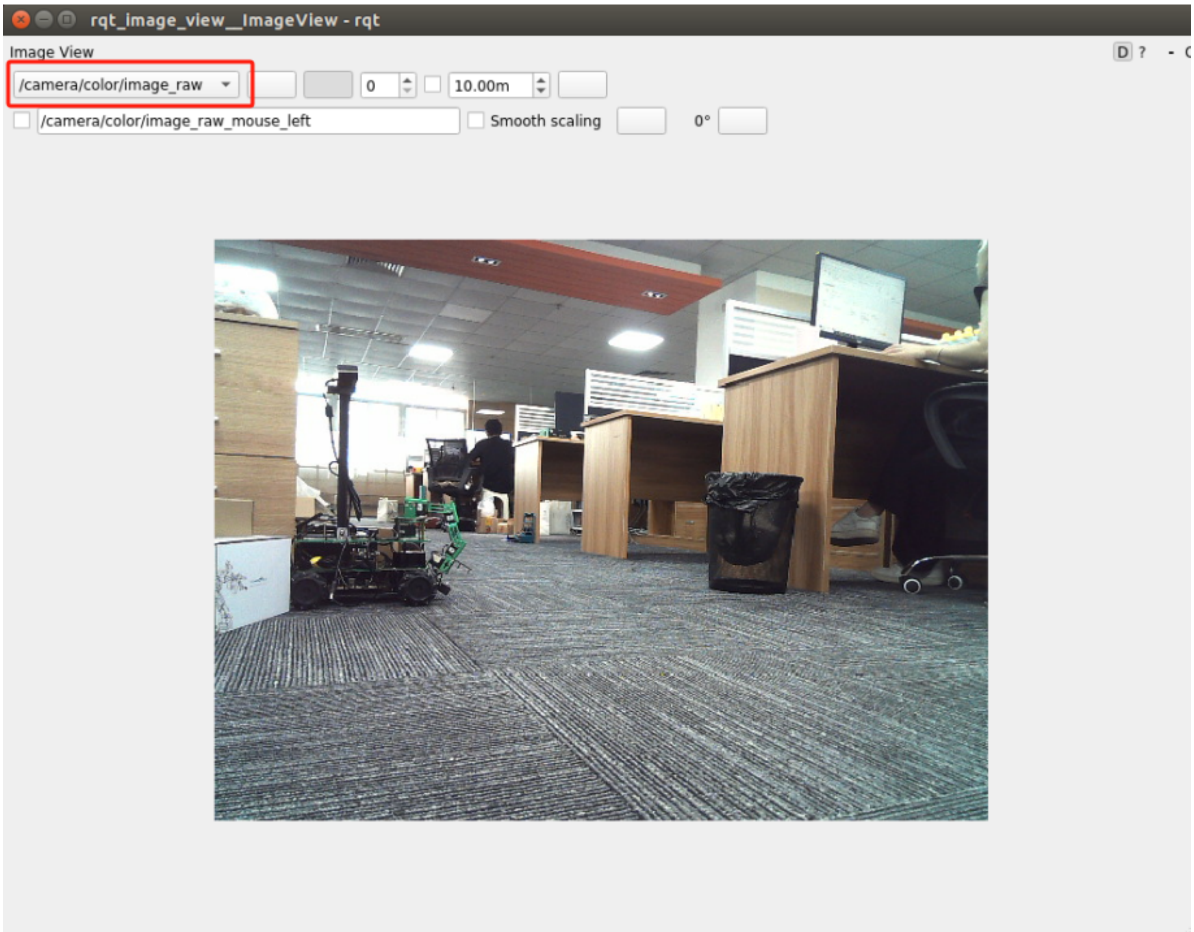
The main topics are as follows:

Topic name	Topic content
/camera/color/image_raw	RGB color image data
/camera/depth/image_raw	Depth depth image data
/camera/depth/points	Depth point cloud data
/camera/ir/image_raw	IR infrared image data

Use the rqt_image_view tool to view image data and enter in the Docker terminal:

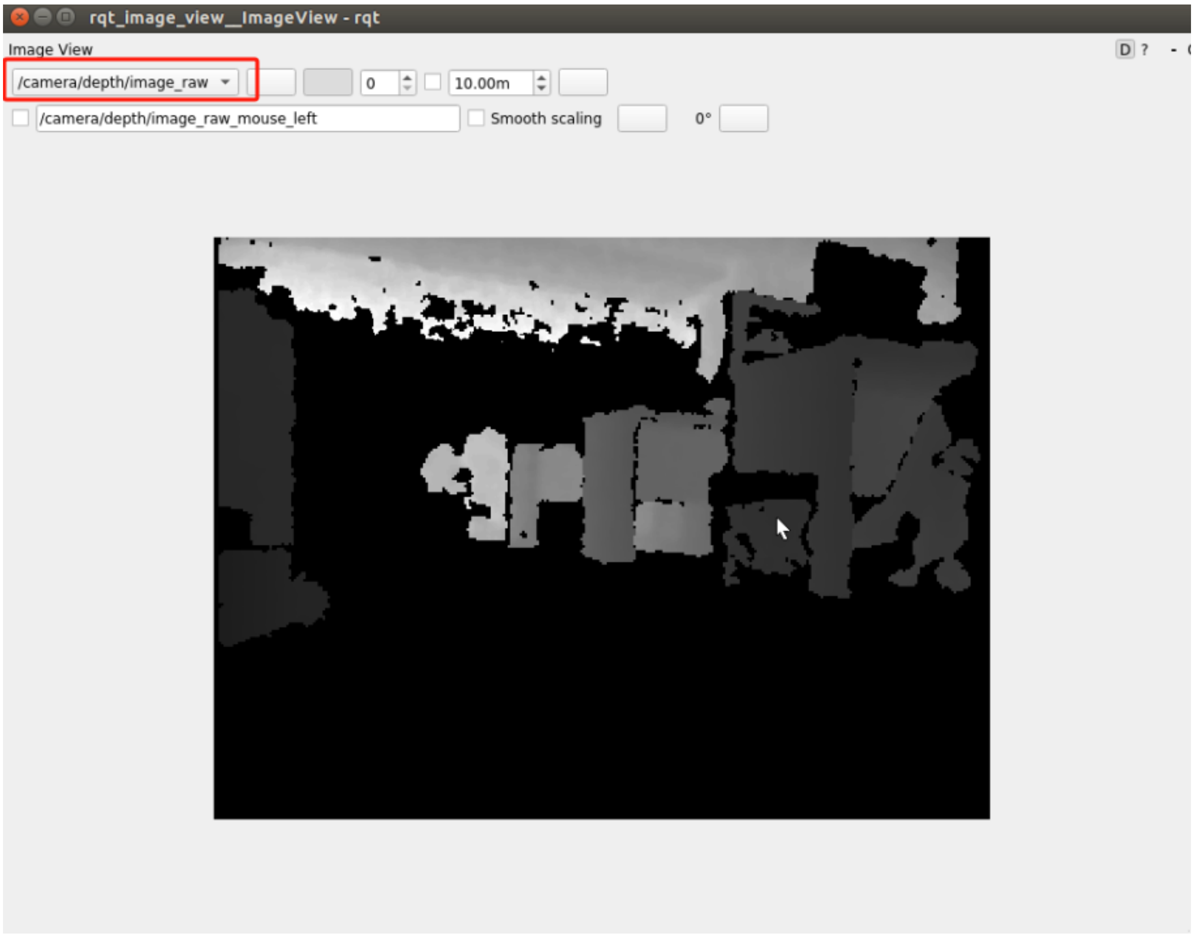
```
ros2 run rqt_image_view rqt_image_view
```

Then select the corresponding image topic that needs to be displayed in the upper left corner to display the RGB image:



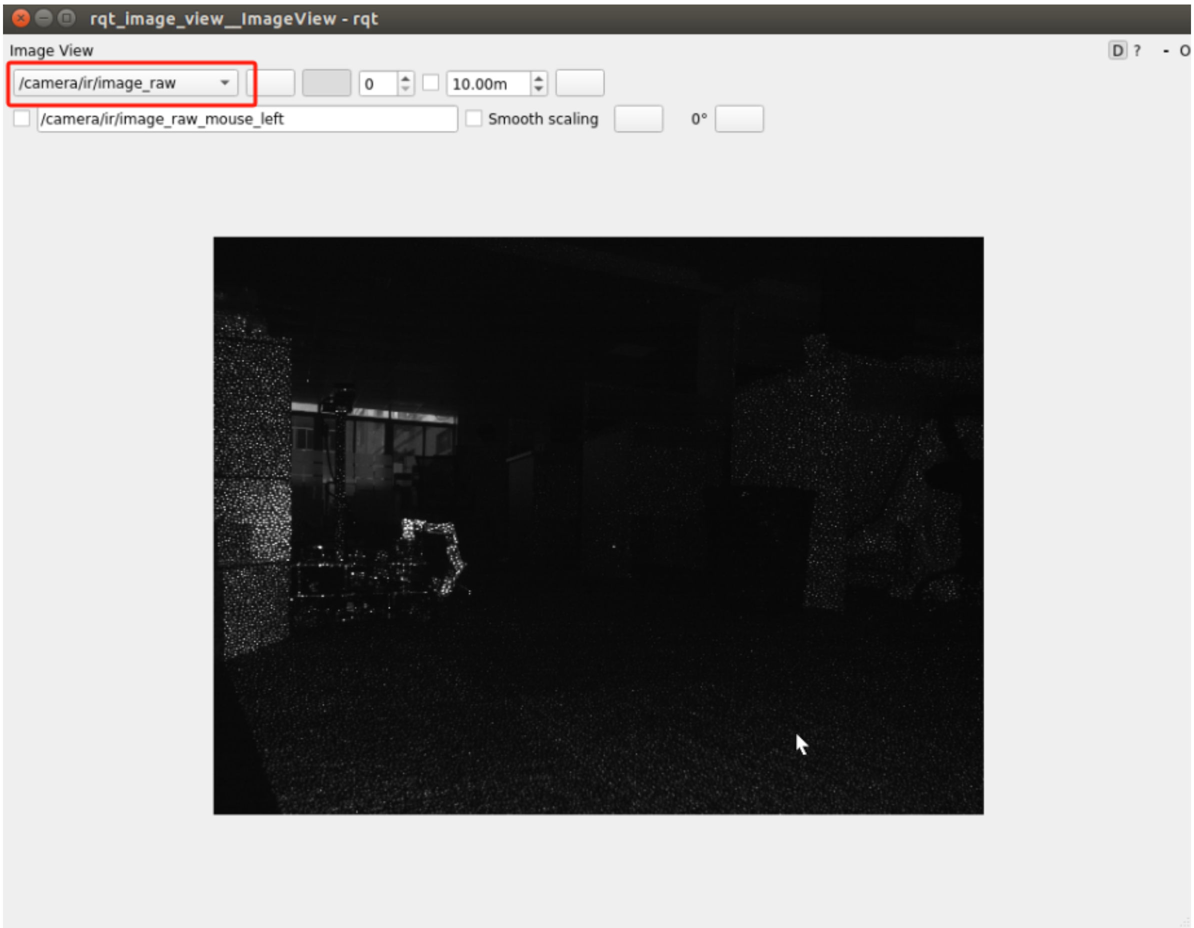
rqt_image_view__ImageView - rqt
Image View
/camera/color/image_raw
0
10.00m
/camera/color/image_raw_mouse_left
Smooth scaling
0°
D ?

Display the depth image:



rqt_image_view__ImageView - rqt
Image View
/camera/depth/image_raw
0
10.00m
/camera/depth/image_raw_mouse_left
Smooth scaling
0°
D ?

Display infrared image:



/camera/ir/image_raw

0

10.00m

☐ /camera/ir/image_raw_mouse_left

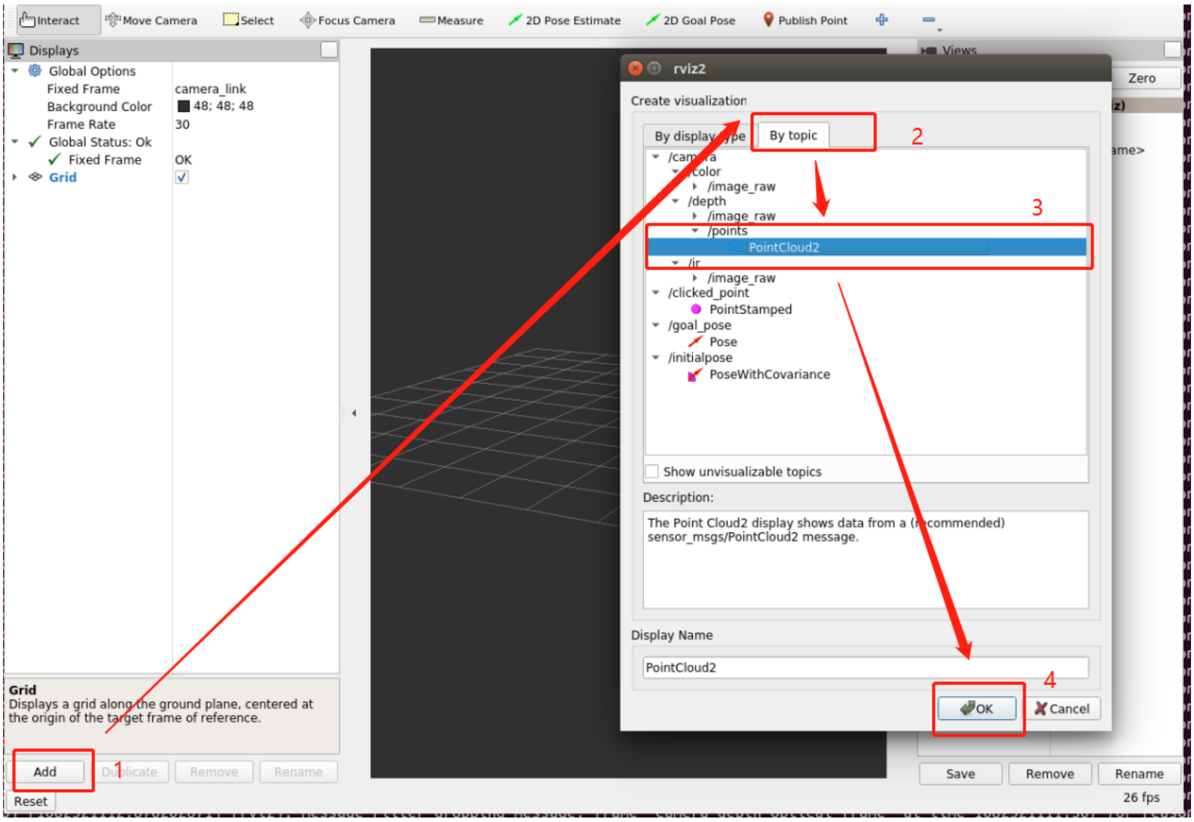
☐ Smooth scaling

0°

Use rviz2 to display the depth point cloud, enter in the Docker terminal:

```
rviz2
```

Then add depth point cloud information in rviz,



Global Options

Fixed Frame camera_link

Background Color ■ 48; 48; 48

Frame Rate 30

✓ Global Status: Ok

✓ Fixed Frame OK

Grid ☑

Grid

Displays a grid along the ground plane, centered at the origin of the target frame of reference.

Create visualization

By display type | By topic

```
/camera
  /color
    /image_raw
  /depth
    /image_raw
    /points
      PointCloud2
  /ir
    /image_raw
/clicked_point
  PointStamped
/goal_pose
  Pose
/initialpose
  PoseWithCovariance
```

☐ Show unvisualizable topics

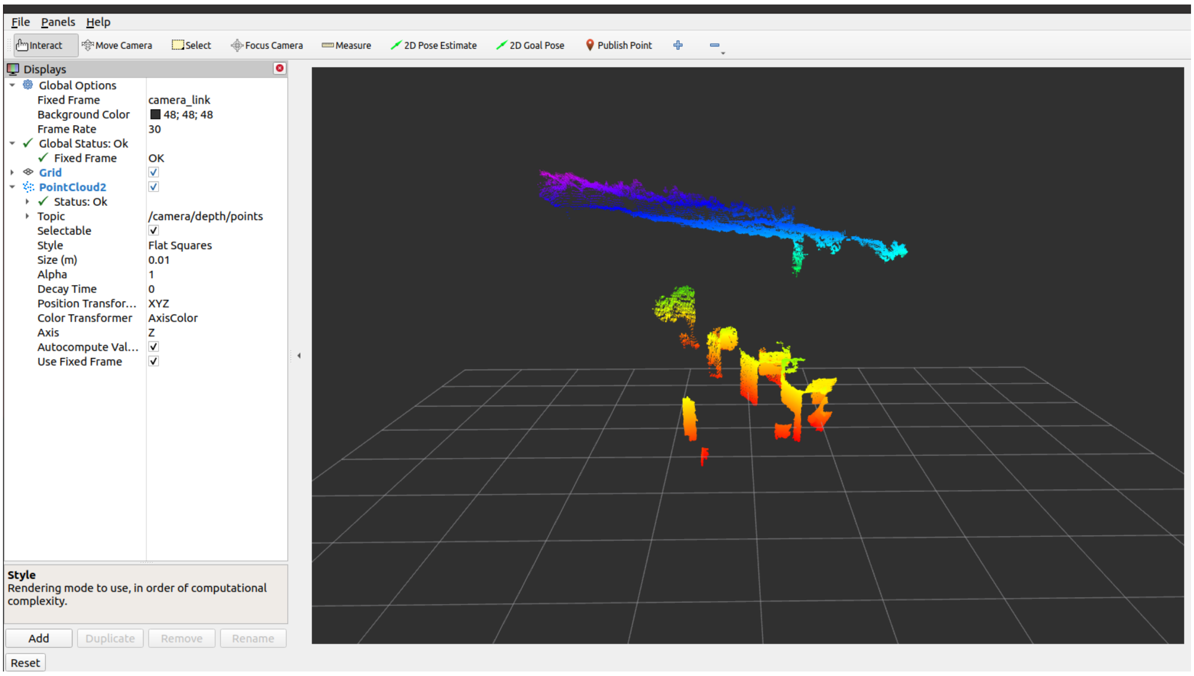
Description:

The Point Cloud2 display shows data from a (recommended) sensor_msgs/PointCloud2 message.

Display Name

PointCloud2

Finally, change **Fixed Frame** to **camera_link** to view the point cloud information:



Global Options	
Fixed Frame	camera_link
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	☑
PointCloud2	☑
✓ Status: Ok	
Topic	/camera/depth/points
Selectable	☑
Style	Flat Squares
Size (m)	0.01
Alpha	1
Decay Time	0
Position Transfor...	XYZ
Color Transformer	AxisColor
Axis	Z
Autocompute Val...	☑
Use Fixed Frame	☑

Style
Rendering mode to use, in order of computational complexity.

If you only need depth information, Astrapro and Astraproplus only need to start the following commands,

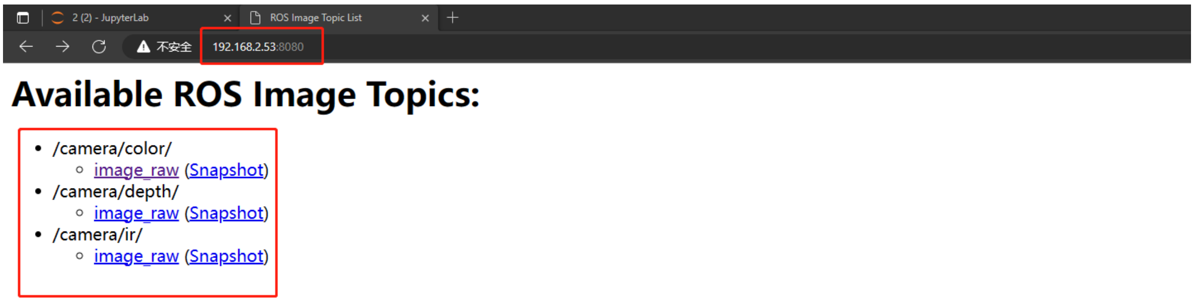
```
ros2 launch astra_camera astra.launch.xml
```

View image data on the web page:

```
ros2 run web_video_server web_video_server
```

Then open the browser, the computer and the host network must be in the same LAN, enter the URL: your host ip+8080,For example, my host network IP is 192.168.2.53, and my docke container also uses the host network, so the network IPs of the two are the same, so enter the URL in the browser:

```
http://192.168.2.53:8080/
```

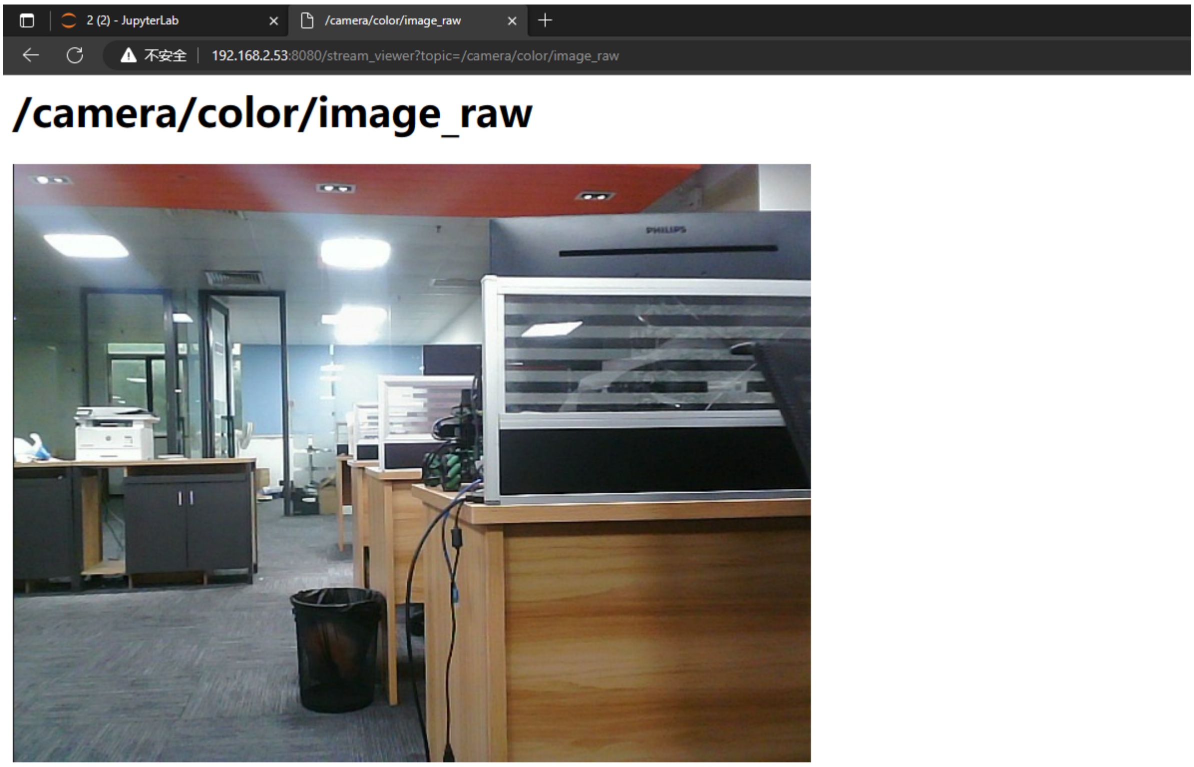


192.168.2.53:8080

Available ROS Image Topics:

- /camera/color/
  - image_raw (Snapshot)
- /camera/depth/
  - image_raw (Snapshot)
- /camera/ir/
  - image_raw (Snapshot)

Then, select one of the image topic data, for example, to display an rgb image, then select image_raw under /camera/color/,



192.168.2.53:8080/stream_viewer?topic=/camera/color/image_raw

/camera/color/image_raw

Note: Because the format of the depth map is different from that of rgb, the depth map cannot be displayed. If you need to see the depth map, you can view it through rviz or rqt.

4. Special instructions

The factory configuration of Muto robot is astro_pro_plus camera. If the user switches to astra_pro camera, please change the camera running part in all courses to run astra_pro camera.