

1. Depth camera use

Before driving the depth camera, you need to be able to recognize the astra camera device in the host; when entering the docker container, you need to mount this astra device in order to be in the docker container and recognize the camera. Matching host has been built environment, do not need additional configuration, if the new host, you need to add the rules file. Adding method is very simple, copy the /etc/udev.rules.d/56-orbbec-usb.rules file under the host machine to the /etc/udev.rules.d directory of the new environment, and then reboot once.

1. Program function description

After the program runs, it drives Astra camera, and can get the image information of color RGB, depth Depth, infrared IR, and depth point cloud information.

2. Program code reference path

After entering the docker container, the location of the source code for this function is located at.

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

3. Program launch

The launch startup command.

launch file	Camera type
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

To start the Astrapro camera, for example, after entering the docker container, terminal type, the

```
11 /dev/astra*
```

This step is to check if the Astra camera device is recognized in Docker. The following appears to indicate that the Astra camera is successfully mounted in the Docker container.

```
root@jetson-desktop: ~  
root@jetson-desktop: ~ 80x24  
root@jetson-desktop:~# ll /dev/astra*  
crw-rw-rw- 1 root video 189, 7 Apr 24 06:18 /dev/astra_pro  
crw-rw-rw- 1 root video 189, 5 Apr 24 06:18 /dev/astrauvc  
root@jetson-desktop:~#
```

In the Docker terminal type, the

```
ros2 launch astra_camera astra_pro.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::OBCameraNodeFactory>. 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). Please separate 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] [INFO] [1682320215.453246486] [camera.camera]: init done.  
[astra_camera_node-1] [INFO] [1682320215.453300028] [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: 2bc5/040301/8  
[astra_camera_node-1] [INFO] [1682320215.612499749] [camera.camera]: Device connected: Astra serial number: AC2MC130005  
[astra_camera_node-1] [INFO] [1682320215.612610583] [camera.camera]: starting device  
[astra_camera_node-1] [INFO] [1682320215.670516713] [camera.camera]: set depth video node Resolution :640x480@30Hz  
[astra_camera_node-1] [INFO] [1682320215.671318436] [camera.camera]: set ir video node Resolution :640x480@30Hz  
[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: AC2MC130005  
[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] [INFO] [1682320215.786017044] [camera.camera]: open camera success  
[astra_camera_node-1] [INFO] [1682320215.791607281] [camera.camera]: set depth video node Resolution :640x480@30Hz  
[astra_camera_node-1] [INFO] [1682320215.793952136] [camera.camera]: set ir video node Resolution :640x480@30Hz  
[astra_camera_node-1] [WARN] [1682320215.791607698] [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] [1682320215.810998069] [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 node 640x480@30 Format UVC_FRAME_FORMAT_MJPEG  
[astra_camera_node-1] [INFO] [1682320216.148300096] [camera.camera]: device started.
```

The topic can be viewed using the following command, entered in the Docker terminal, the

```
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:/#
```

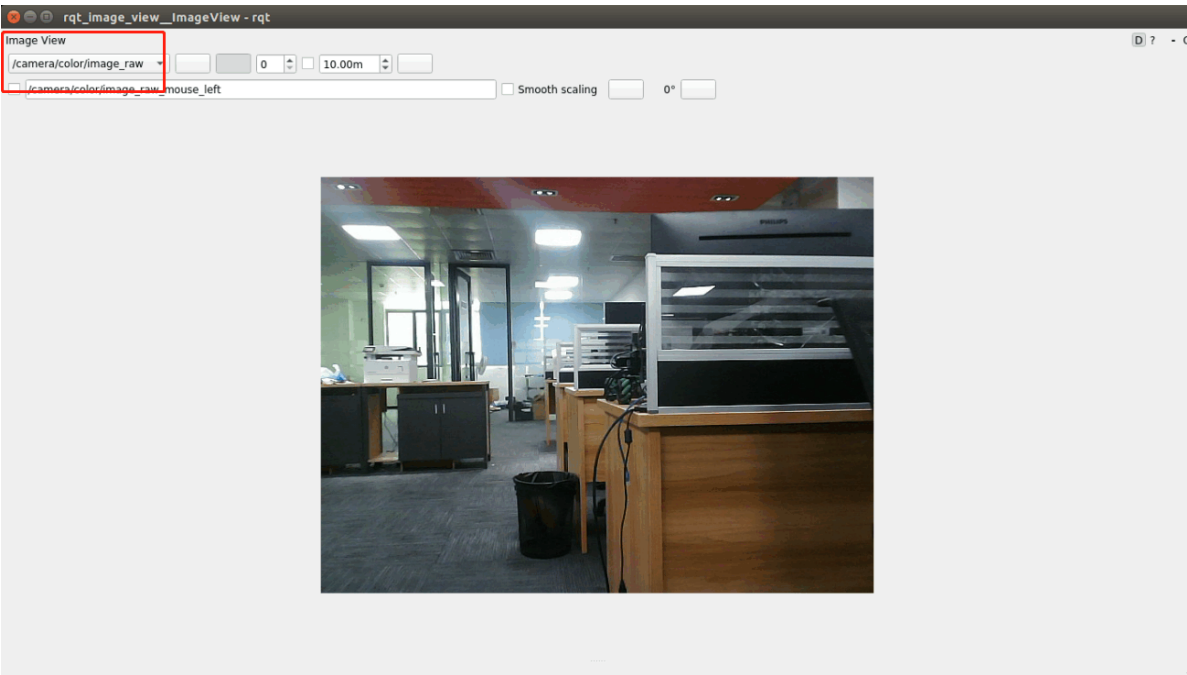
The main topics are as follows.

topic name	Topics
/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 image data

To view the image data using the rqt_image_view tool, type in the Docker terminal, the

```
ros2 run rqt_image_view rqt_image_view
```

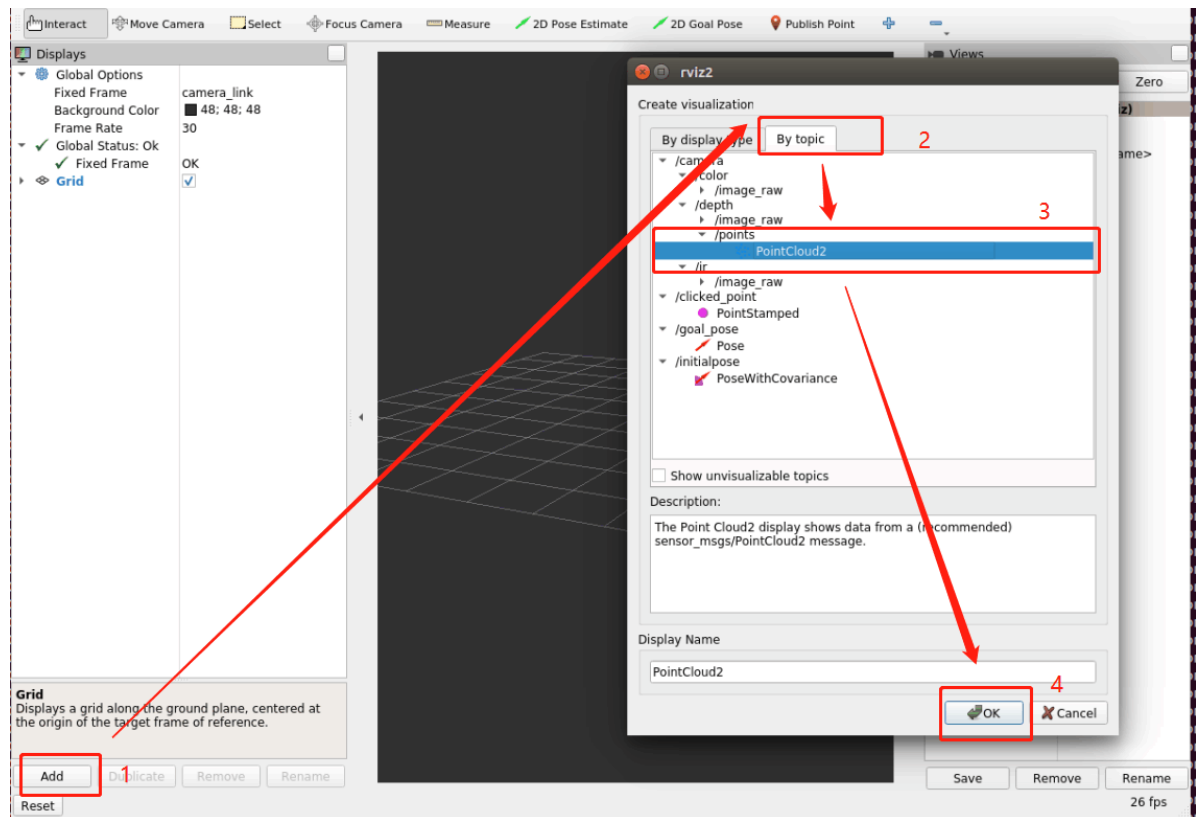
Then in the upper left corner, select the corresponding need to display the image topic, here to display RGB images as an example, the



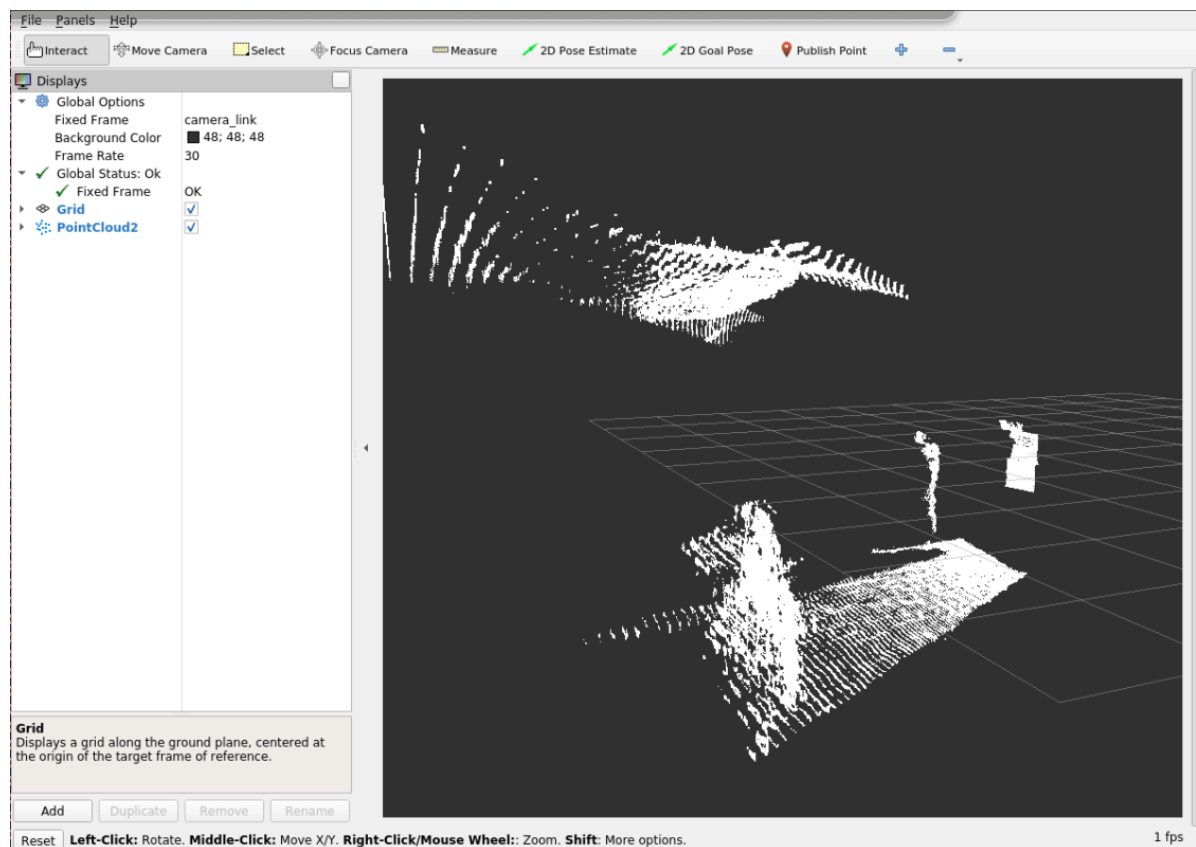
To display the depth point cloud using rviz2, type in the Docker terminal, the

```
rviz2
```

Then add the depth point cloud information in rviz.



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



If only depth information is required, Astrapro and Astraproplus simply start the following command.

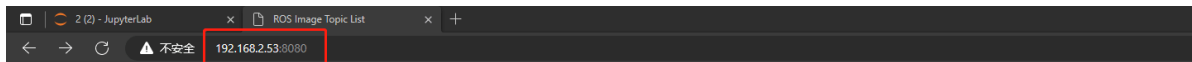
```
ros2 launch astra_camera astra.launch.xml
```

web page to view the image data.

```
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, my docke containers are also using the host network, so the two network ip is the same, so enter the URL in the browser:

```
http://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, e.g. to display an rgb image, then select image_raw under /camera/color/, the

/camera/color/image_raw

