

# Dabai\_DCW2 camera

Before starting this function, you need to close the process of the big program and APP. Enter the following program in the terminal to close the process of the big program and APP.

```
sh ~/app_Arm/kill_YahboomArm.sh
sh ~/app_Arm/stop_app.sh
```

If you need to start the big program and APP again later, start the terminal.

```
sudo systemctl start yahboom_arm.service
sudo systemctl start yahboom_app.service
```

## 1. Function description

After starting the camera, you can get RGB color images, Depth depth images and IR infrared images.

## 2. Startup and operation

### 2.1. Startup

After the camera is connected to the mainboard via a data cable, open the terminal and input,

```
roslaunch orbbec_camera dabai_dcw2.launch
```

As shown in the figure below, if the terminal prints the above content, it means that the camera is running normally and the startup is successful. You can view the topics published by the camera node through the following command, and enter the end point,

```
process[camera/camera-1]: started with pid [6946]
[09/23 15:07:50.252650][info][6946][Context.cpp:67] Context created with config: /home/jetson/dofbot_pro_ws/src/OrbbecSDK_ROS/config/OrbbecSDKConfig_v1.0.xml
[09/23 15:07:50.252707][info][6946][Context.cpp:72] Context work_dir=/home/jetson/.ros
[09/23 15:07:50.333755][warning][6946][OpenNIDeviceInfo.cpp:180] New openni device matched.
[09/23 15:07:50.333950][info][6946][LinuxPal.cpp:110] Create PollingDeviceWatcher!
[09/23 15:07:50.334125][info][6946][DeviceManager.cpp:15] Current found device(s): (1)
[09/23 15:07:50.334157][info][6946][DeviceManager.cpp:24]   Name: DaBai DCW2, PID: 0x06a0, SN/ID: AUCI14P00E2, Connection: USB2.0
[ INFO ] [1727075270.351788249]: queryDevice: first query device
[ INFO ] [1727075270.355597959]: deviceConnectCallback : deviceConnectCallback start
[ INFO ] [1727075270.455850720]: deviceConnectCallback : Before process lock lock
[ INFO ] [1727075270.455964619]: deviceConnectCallback : After process lock lock
[ INFO ] [1727075270.456025449]: deviceConnectCallback : selectDevice start
[ INFO ] [1727075270.456098778]: Connecting to the default device
[ INFO ] [1727075270.528395834]: deviceConnectCallback : selectDevice end
[ INFO ] [1727075270.706238654]: stream depth is enabled - width: 640, height: 400, fps: 10, Format: Y11
[ INFO ] [1727075270.707099069]: stream ir is enabled - width: 640, height: 400, fps: 10, Format: Y10
[ INFO ] [1727075270.708218270]: stream color is enabled - width: 640, height: 480, fps: 10, Format: MJPG
[ INFO ] [1727075270.731815914]: set align mode: ALIGN_D2C_HW_MODE
[ INFO ] [1727075271.456503381]: Create color frame read thread.
[ WARN ] [1727075272.380005936]: Publishing dynamic camera transforms (/tf) at 10 Hz
[ INFO ] [1727075272.446413316]: Stream left_ir is disabled.
[ INFO ] [1727075272.446496384]: Stream right_ir is disabled.
[ INFO ] [1727075272.482674425]: stream depth exposure 8000
[ INFO ] [1727075272.483313032]: stream ir exposure 8000
[ INFO ] [1727075272.483634159]: stream color exposure 100
[ INFO ] [1727075272.484492389]: stream depth gain 1388
[ INFO ] [1727075272.485062980]: stream ir gain 1388
[ INFO ] [1727075272.485420190]: stream color gain 0
[ WARN ] [1727075272.497436674]: get white balance error Property is not supported! propertyId: 2004
[ INFO ] [1727075272.497554471]: Device DaBai DCW2 connected
[ INFO ] [1727075272.497623946]: Serial number: AUCI14P00E2
[ INFO ] [1727075272.497677015]: Firmware version: RD1014
[ INFO ] [1727075272.497737376]: Hardware version:
[ INFO ] [1727075272.497782738]: device uid: 1-2.4.1-9
[ INFO ] [1727075272.497834194]: deviceConnectCallback : deviceConnectCallback end
```

```
roslaunch orbbec_camera dabai_dcw2.launch
```

As shown in the figure below, the published color image topic is **/camera/color/image\_raw**, the published depth image topic is **/camera/depth/image\_raw**, the published infrared image topic is **/camera/ir/image\_raw**, and the published point cloud data topic is **/camera/depth/points**.

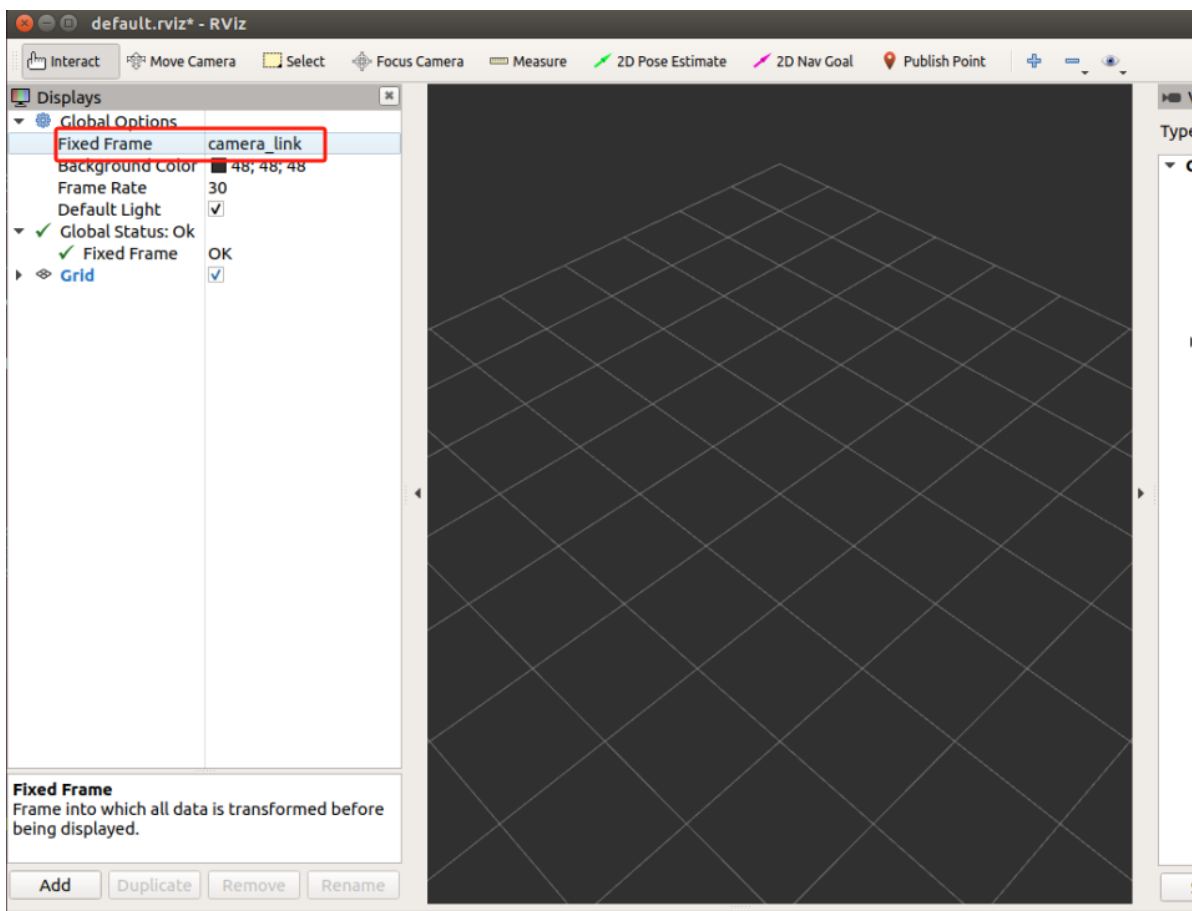
```
jetson@yahboom:~$ rostopic info /camera/camera
-----
Node [/camera/camera]
Publications:
 * /camera/color/camera_info [sensor_msgs/CameraInfo]
 * /camera/color/image_raw [sensor_msgs/Image]
 * /camera/depth/camera_info [sensor_msgs/CameraInfo]
 * /camera/depth/image_raw [sensor_msgs/Image]
 * /camera/depth/points [sensor_msgs/PointCloud2]
 * /camera/depth_registered/points [sensor_msgs/PointCloud2]
 * /camera/ir/camera_info [sensor_msgs/CameraInfo]
 * /camera/ir/image_raw [sensor_msgs/Image]
 * /rosout [rosgraph_msgs/Log]
 * /tf [tf2_msgs/TFMessage]
 * /tf_static [tf2_msgs/TFMessage]
```

## 2.2, Operation

Use rviz to view the three images and point cloud provided by the camera, restart a terminal input,

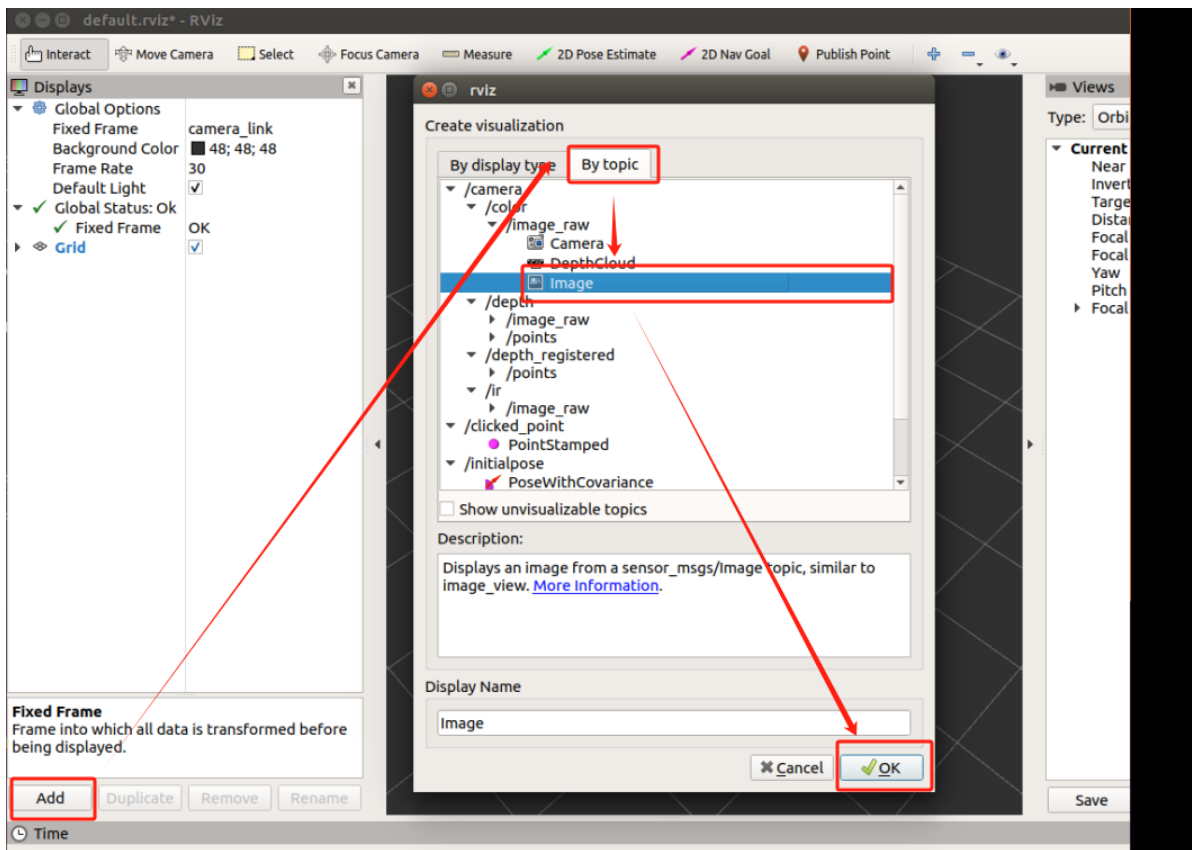
```
rviz
```

After rviz is started, set rviz according to the following diagram,

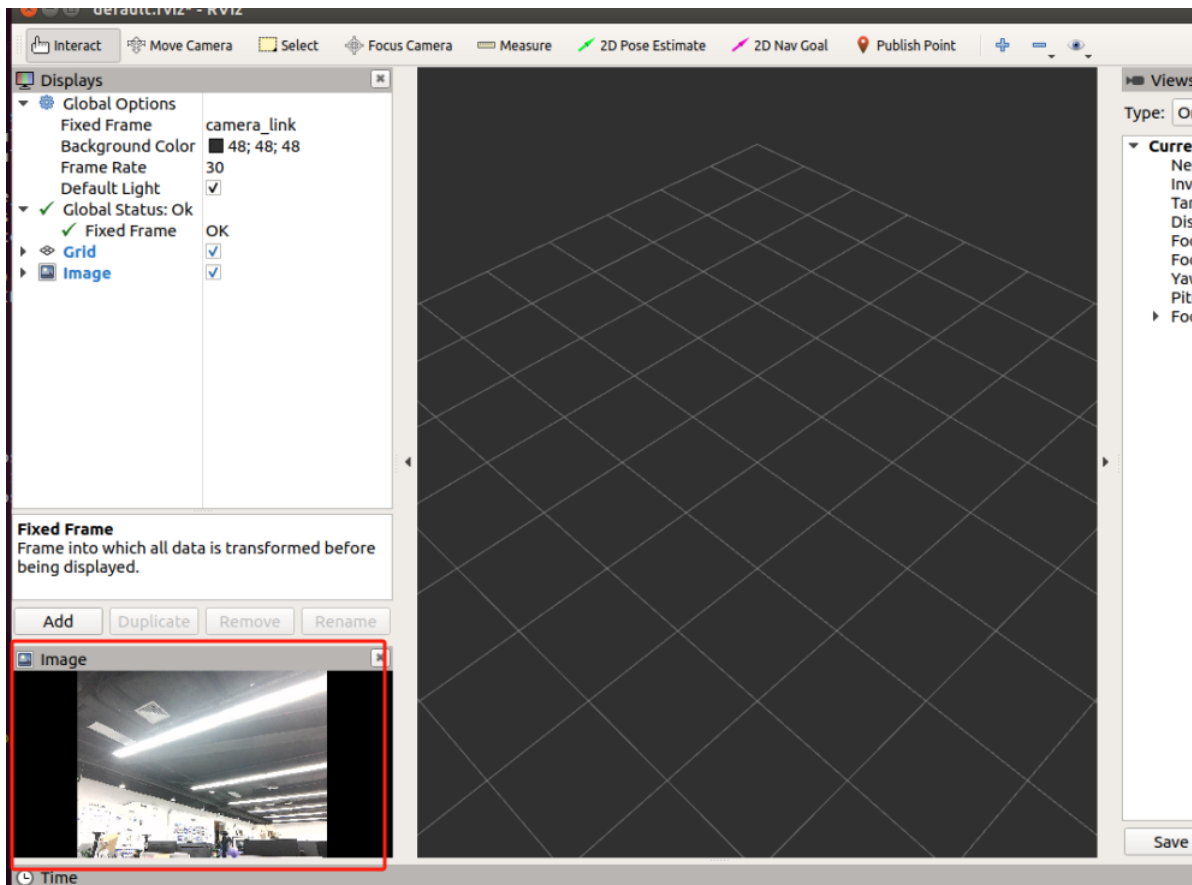


Change [Fixed Frame] here to **camera\_link**

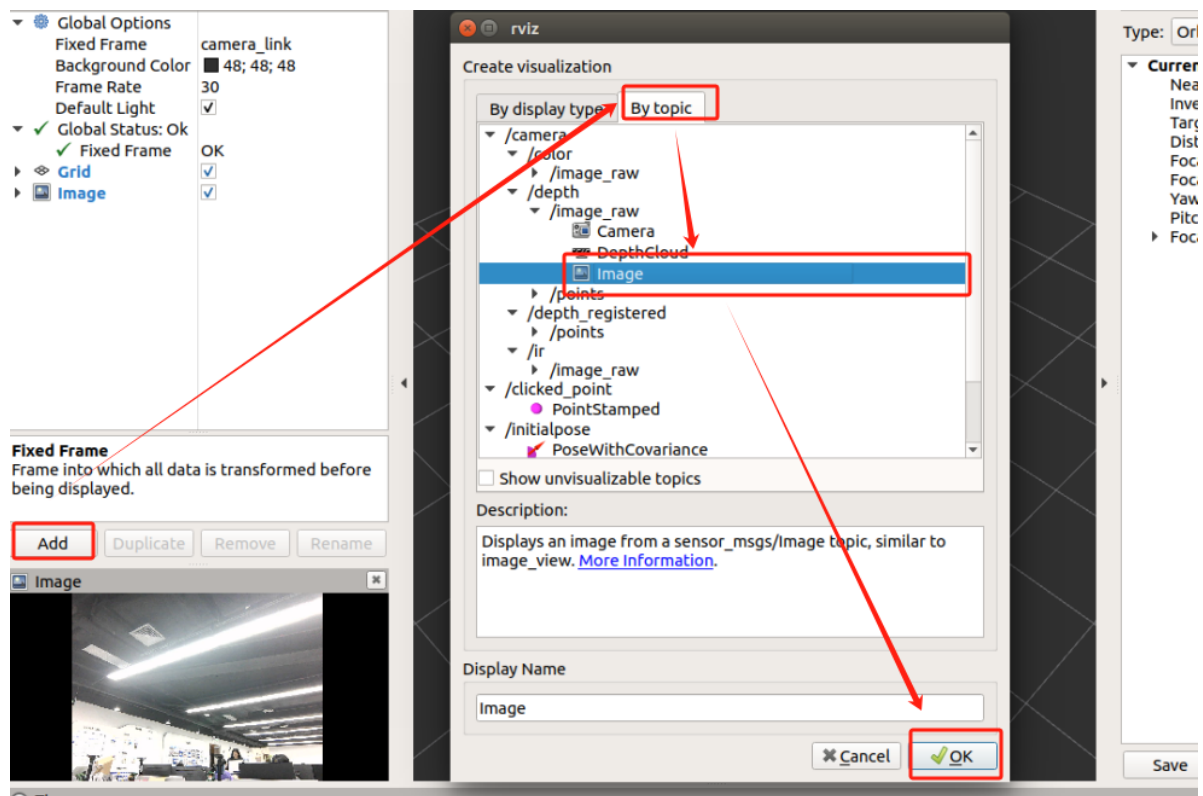
Add color image display



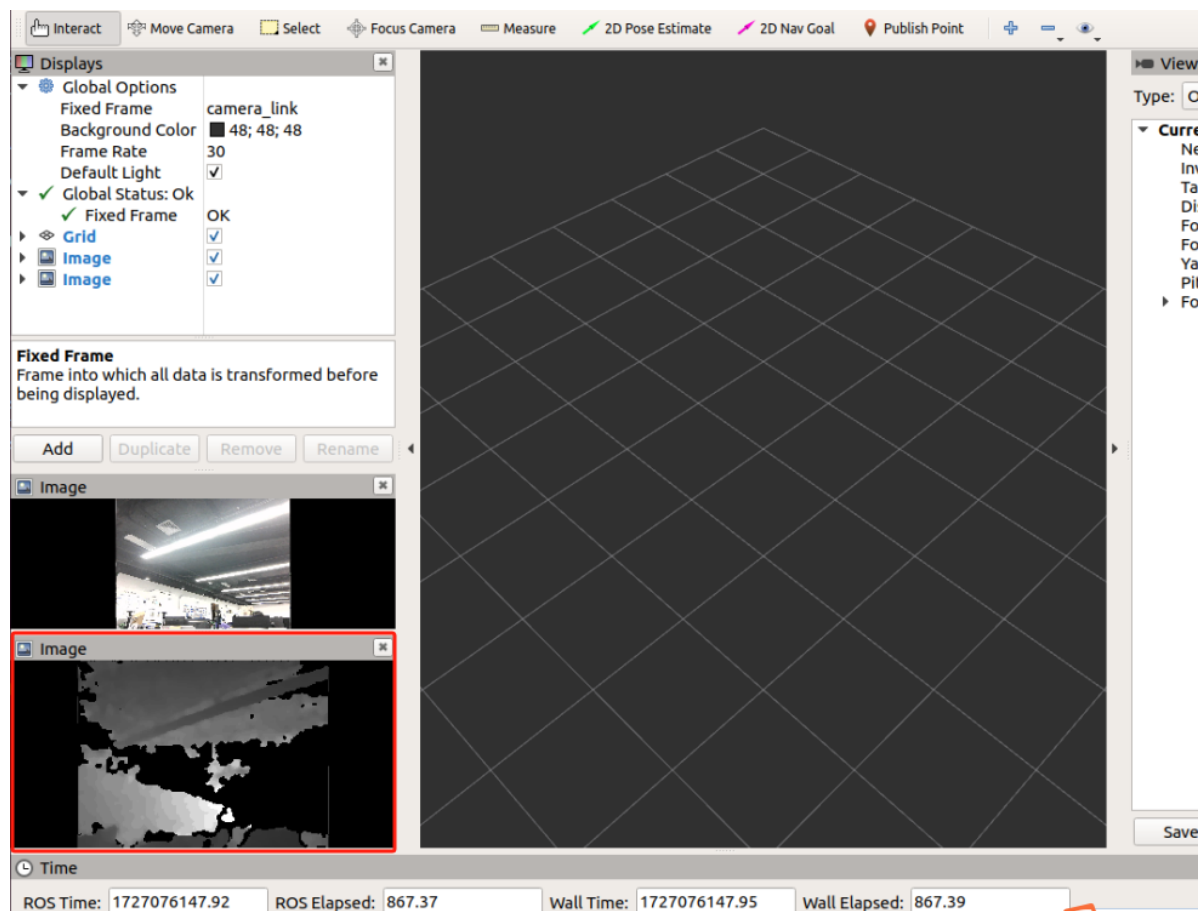
Click [Add], then click [By topic] , select [/color/image\_raw/Image], and finally click [ok] to confirm.  
The color image is as follows.



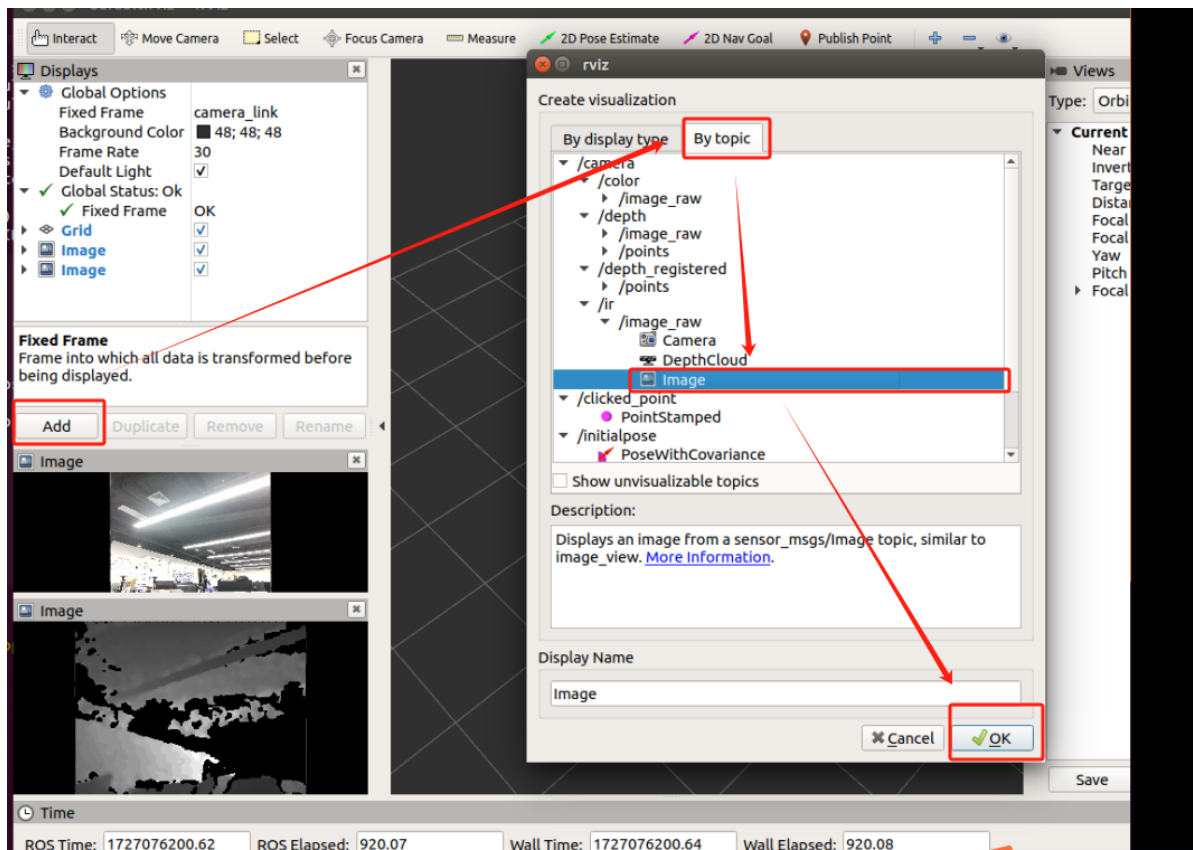
Add depth image display



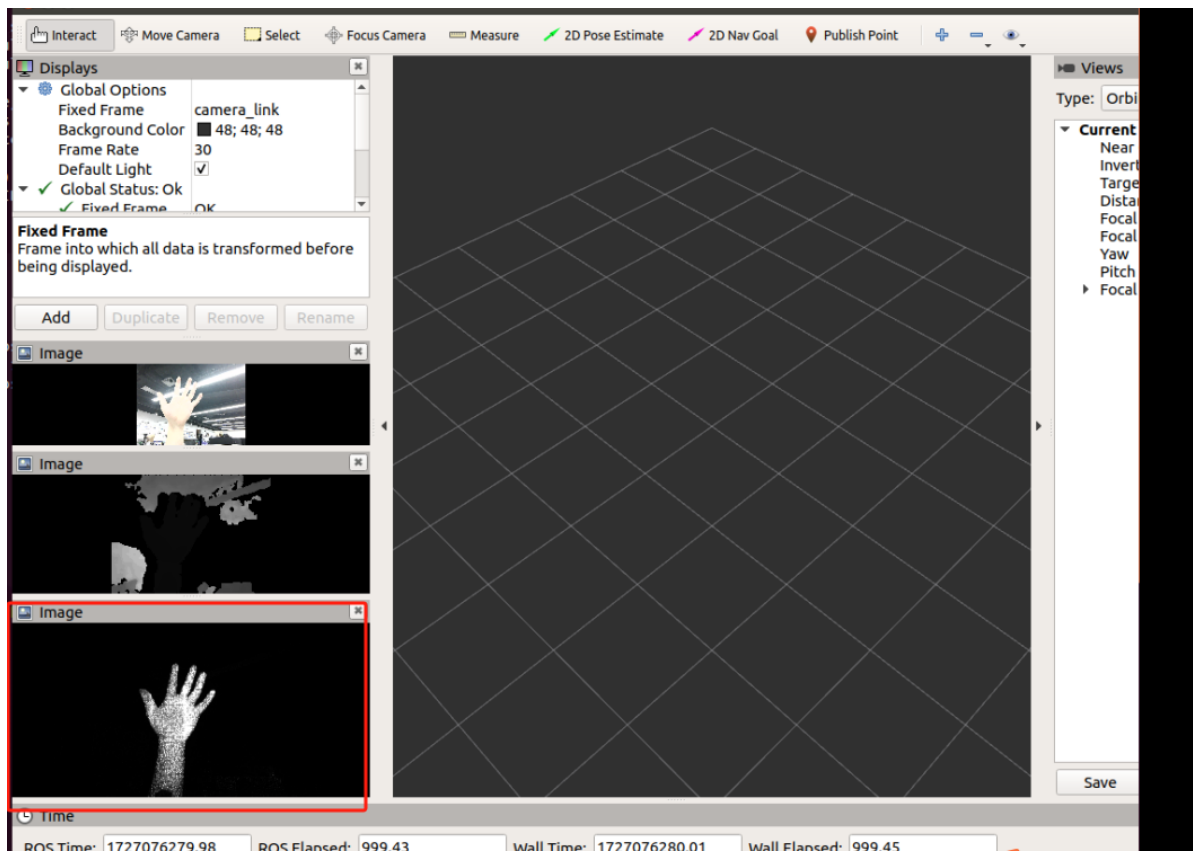
Click [Add], then click [By topic] , select [/depth/image\_raw/Image], and finally click [ok] to confirm. The depth color image is as follows.



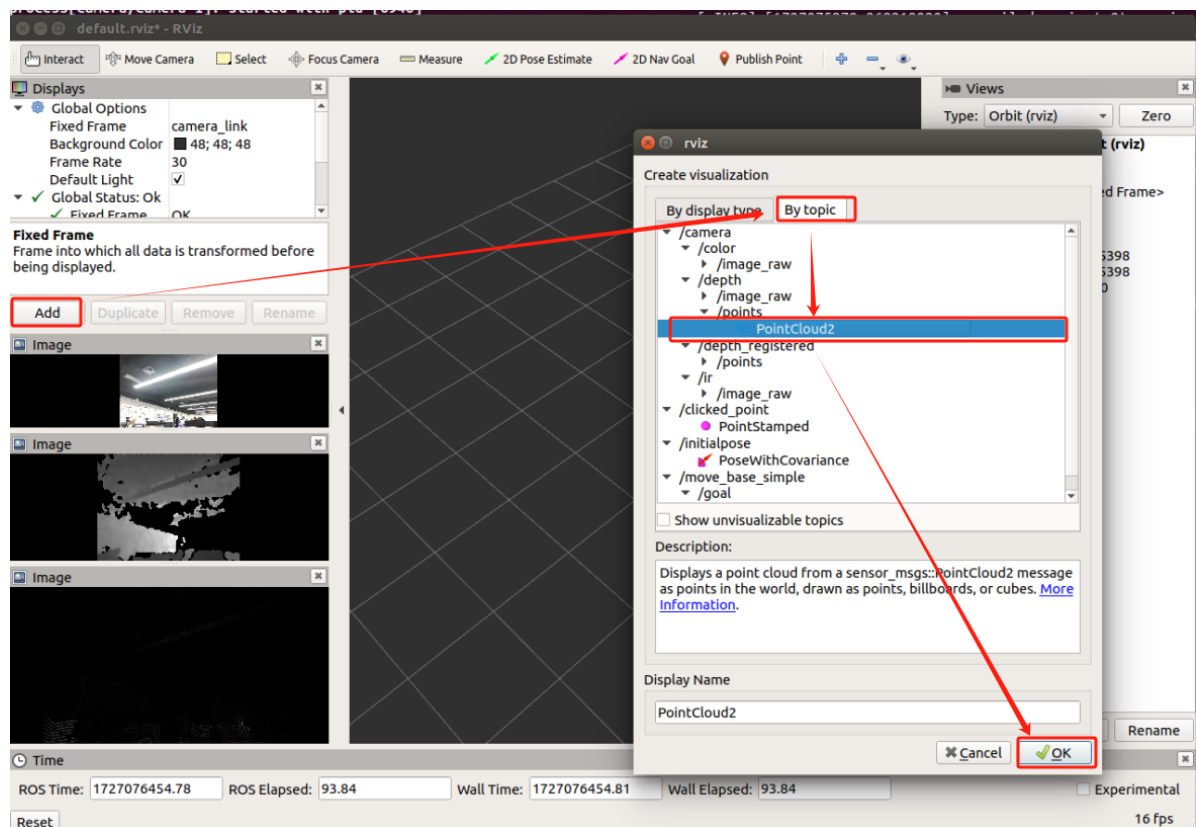
Add infrared image display



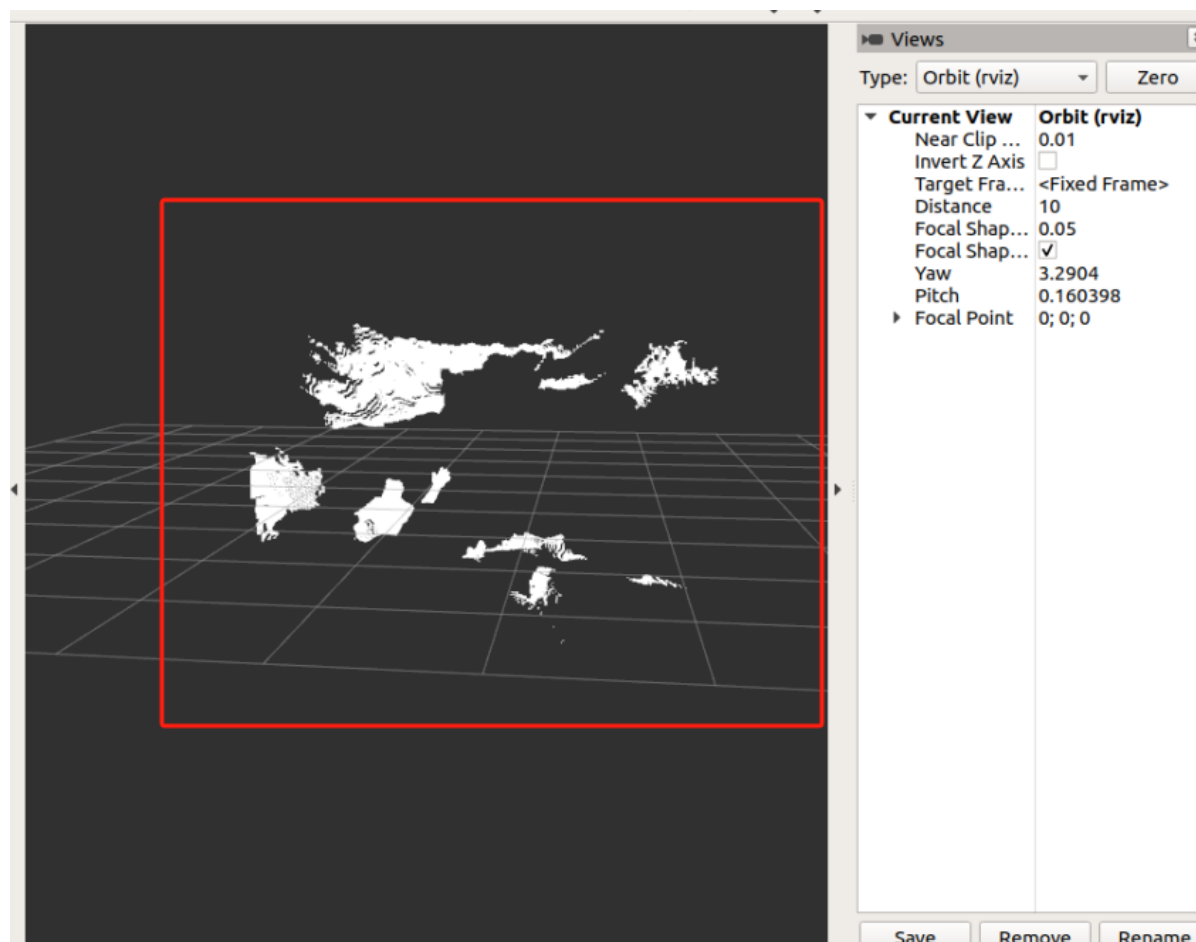
Click [Add], then click [By topic] , select [/ir/image\_raw/Image], and finally click [ok] to confirm.  
The infrared image is as follows.



View the depth point cloud data

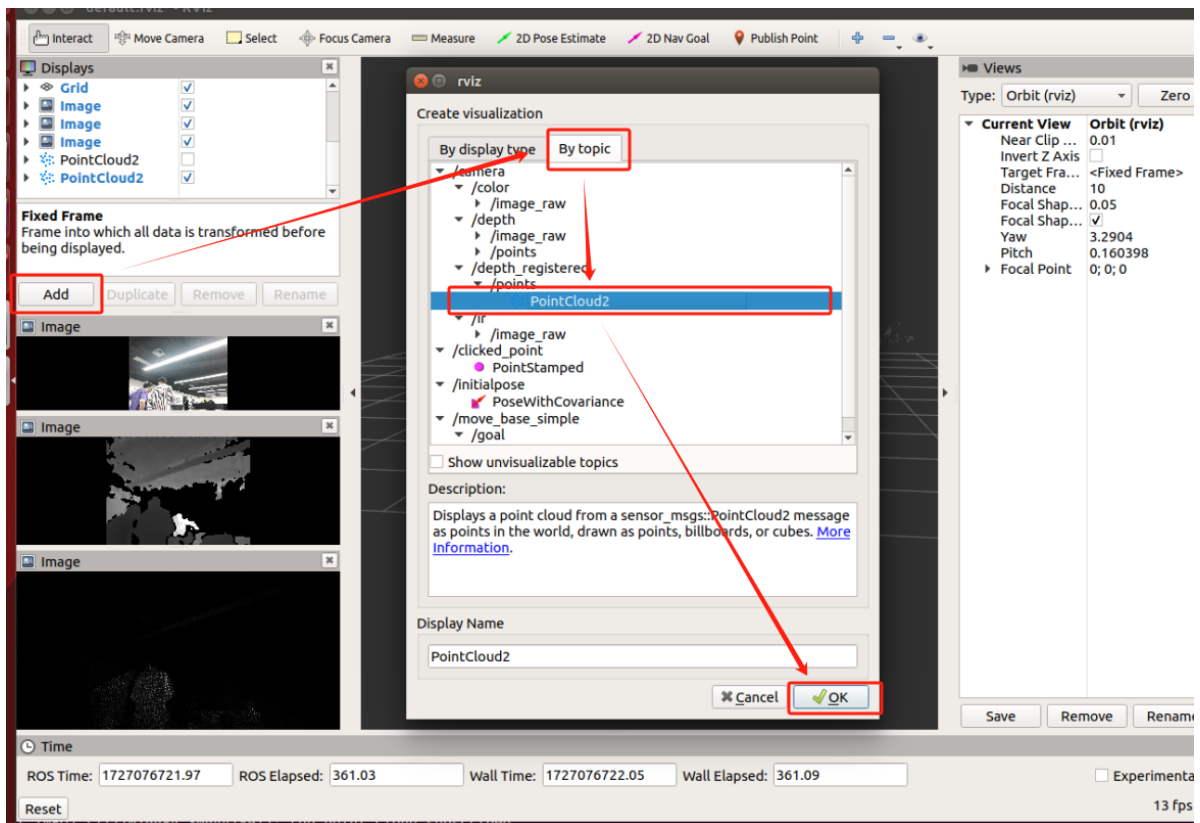


Click [Add], then click [By topic] , select [/depth/points/PointCloud2], and finally click [ok] to confirm. The depth point cloud image is as follows.

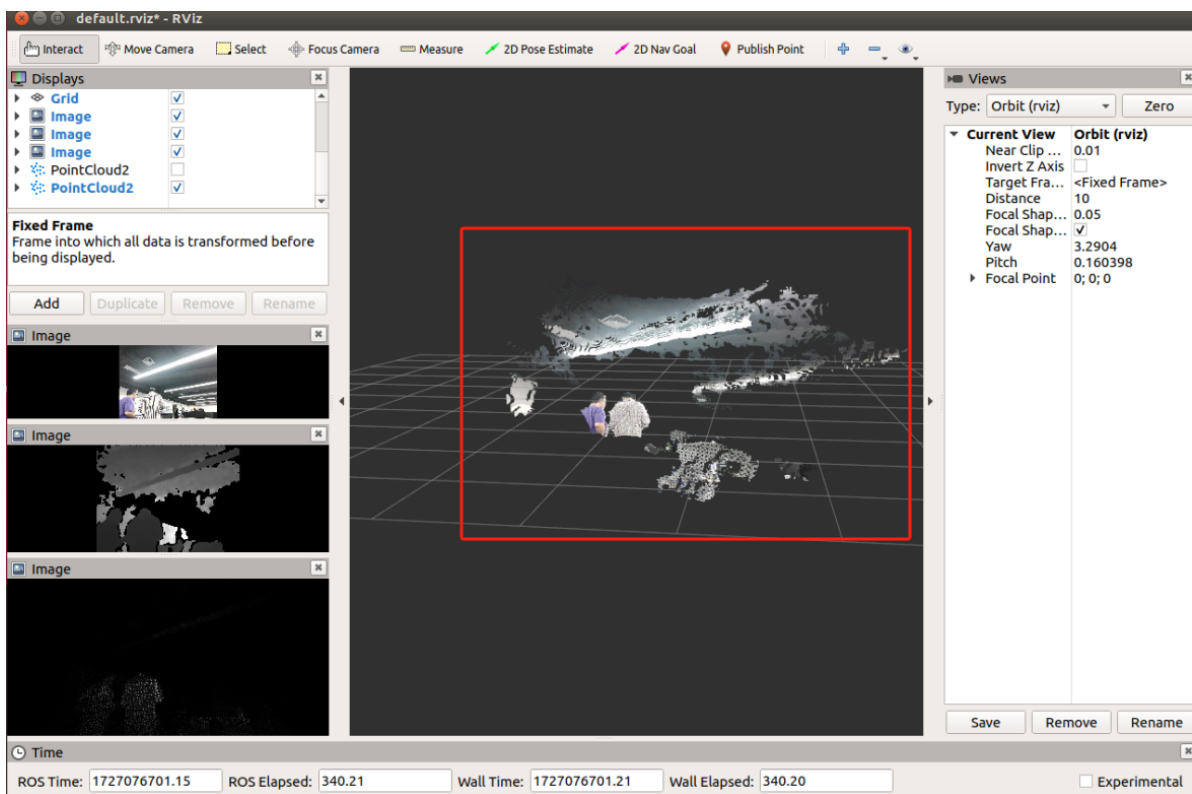


View the color point cloud data





Click [Add], then click [By topic], select [/depth\_registered/points/PointCloud2], and finally click [ok] to confirm. The color point cloud image is as follows.



### 3. Camera parameters

### 3.1. Hardware parameter table

DaBai DCW2/DW2 Depth Camera					
Basic parameters	Parameters		Specifications	Remark	
	Name		DaBai DCW2/DW2		
	Model		DaBai DCW2: G10254-001 DaBai DW2: G11254-001		
	Working distance		Normal energy mode: 0.15-3m High energy mode: 0.15-5m		
	Device size		89.82mm*25.1mm*25.1mm		
	Power consumption		DaBai DCW2: Peak power consumption <7W; average power consumption <2.5W DaBai DW2: Peak power consumption <6W; average power consumption <2.0W		
	Baseline		40mm		
	Interface type		USB Type-C		
	Communication/power supply mode		USB 2.0		
	Working temperature		Normal energy mode: -10℃-50℃ High energy mode: -10℃-40℃		
	Working humidity		5% - 95%RH		
	Storage temperature		-20℃ - 70℃		
	RE		-6dB@10m, Class B		Comes with 3m cable
	ESD		8k/15k Class A		Comes with 3m cable
	Relative accuracy		<1%@1m; <1.3%@2m		

### 3.2. Camera parameter acquisition and setting

The initialization parameters of the camera have been set in the launch file. The launch file path is `/home/jetson/dofbot_pro_ws/src/orbbecSDK_ROS/launch/dabai_dcw2.launch`

The contents are as follows,,

```
<launch>
  <!-- unique camera name-->
  <arg name="camera_name" default="camera" />
  <!-- Hardware depth registration -->
  <arg name="depth_registration" default="false" />
  <arg name="serial_number" default="" />
  <arg name="usb_port" default="" />
  <arg name="device_num" default="1" />
  <arg name="vendor_id" default="0x2bc5" />
  <arg name="product_id" default="" />
  <arg name="enable_point_cloud" default="true" />
  <arg name="enable_colored_point_cloud" default="true" />
  <arg name="connection_delay" default="100" />
  <arg name="color_width" default="640" />
  <arg name="color_height" default="480" />
  <arg name="color_fps" default="10" />
  <arg name="enable_color" default="true" />
  <arg name="color_format" default="MJPG" />
  <arg name="flip_color" default="false" />
  <arg name="enable_color_auto_exposure" default="false" />
  <arg name="depth_width" default="640" />
  <arg name="depth_height" default="400" />
  <arg name="depth_fps" default="10" />
  <arg name="enable_depth" default="true" />
  <arg name="depth_format" default="Y11" />
  <arg name="flip_depth" default="false" />
  <arg name="ir_width" default="640" />
  <arg name="ir_height" default="400" />
  <arg name="ir_fps" default="10" />
  <arg name="enable_ir" default="true" />
```



```

<arg name="ir_format" default="Y10" />
<arg name="flip_ir" default="false" />
<arg name="enable_ir_auto_exposure" default="true" />
<arg name="publish_tf" default="true" />
<arg name="tf_publish_rate" default="10.0" />
<arg name="ir_info_uri" default="" />
<arg name="color_info_uri" default="" />
<arg name="log_level" default="none" />
<arg name="enable_d2c_viewer" default="false" />
<arg name="enable_pipeline" default="true" />
<arg name="enable_soft_filter" default="true" />
<group ns="$(arg camera_name)">
  <node name="camera" pkg="orbbeccamera" type="orbbeccamera_node"
output="screen">
  <param name="camera_name" value="$(arg camera_name)" />
  <param name="depth_registration" value="$(arg depth_registration)" />
  <param name="serial_number" type="string" value="$(arg serial_number)" />
  <param name="usb_port" type="string" value="$(arg usb_port)" />
  <param name="vendor_id" value="$(arg vendor_id)" />
  <param name="product_id" value="$(arg product_id)" />
  <param name="enable_point_cloud" value="$(arg enable_point_cloud)" />
  <param name="enable_colored_point_cloud" value="$(arg
enable_colored_point_cloud)" />
  <param name="connection_delay" value="$(arg connection_delay)" />
  <param name="color_width" value="$(arg color_width)" />
  <param name="color_height" value="$(arg color_height)" />
  <param name="color_fps" value="$(arg color_fps)" />
  <param name="enable_color" value="$(arg enable_color)" />
  <param name="color_format" value="$(arg color_format)" />
  <param name="flip_color" value="$(arg flip_color)" />
  <param name="enable_color_auto_exposure" value="$(arg
enable_color_auto_exposure)" />
  <param name="depth_width" value="$(arg depth_width)" />
  <param name="depth_height" value="$(arg depth_height)" />
  <param name="depth_fps" value="$(arg depth_fps)" />
  <param name="enable_depth" value="$(arg enable_depth)" />
  <param name="depth_format" value="$(arg depth_format)" />
  <param name="flip_depth" value="$(arg flip_depth)" />
  <param name="ir_width" value="$(arg ir_width)" />
  <param name="ir_height" value="$(arg ir_height)" />
  <param name="ir_fps" value="$(arg ir_fps)" />
  <param name="enable_ir" value="$(arg enable_ir)" />
  <param name="ir_format" value="$(arg ir_format)" />
  <param name="flip_ir" value="$(arg flip_ir)" />
  <param name="enable_ir_auto_exposure" value="$(arg
enable_ir_auto_exposure)" />
  <param name="publish_tf" value="$(arg publish_tf)" />
  <param name="tf_publish_rate" value="$(arg tf_publish_rate)" />
  <param name="ir_info_uri" value="$(arg ir_info_uri)" />
  <param name="color_info_uri" value="$(arg color_info_uri)" />
  <param name="log_level" value="$(arg log_level)" />
  <param name="enable_d2c_viewer" value="$(arg enable_d2c_viewer)" />
  <param name="enable_pipeline" value="$(arg enable_pipeline)" />
  <param name="device_num" value="$(arg device_num)" />
  <param name="enable_soft_filter" value="$(arg enable_soft_filter)" />
  <remap from="/$(arg camera_name)/depth/color/points"
to="/$(arg camera_name)/depth_registered/points" />
</node>

```

```
</group>
</launch>
```

After the camera is running, we can use some services to get or set some parameters, such as getting exposure value and setting exposure value. First, check which services can be called by the camera node, and enter in the terminal,

```
rostopic info /camera/camera
```

As shown in the figure below, after the camera node is started, some services are provided to us,

```
Services:
* /camera/camera/get_loggers
* /camera/camera/set_logger_level
* /camera/get_auto_white_balance
* /camera/get_camera_params
* /camera/get_color_auto_exposure
* /camera/get_color_camera_info
* /camera/get_color_exposure
* /camera/get_color_gain
* /camera/get_depth_auto_exposure
* /camera/get_depth_camera_info
* /camera/get_depth_exposure
* /camera/get_depth_gain
* /camera/get_device_info
* /camera/get_device_type
* /camera/get_ir_auto_exposure
* /camera/get_ir_camera_info
* /camera/get_ir_exposure
* /camera/get_ir_gain
* /camera/get_ldp_status
* /camera/get_sdk_version
* /camera/get_serial
* /camera/get_white_balance
* /camera/reset_color_exposure
* /camera/reset_color_gain
* /camera/reset_depth_exposure
* /camera/reset_depth_gain
* /camera/reset_ir_exposure
* /camera/reset_ir_gain
* /camera/reset_white_balance
* /camera/save_images
* /camera/save_point_cloud
* /camera/set_auto_white_balance
* /camera/set_color_auto_exposure
* /camera/set_color_exposure
* /camera/set_color_gain
* /camera/set_color_mirror
* /camera/set_depth_auto_exposure
* /camera/set_depth_exposure
* /camera/set_depth_gain
* /camera/set_depth_mirror
* /camera/set_fan_work_mode
* /camera/set_floor
```

### 3.2.1, Camera parameter acquisition

Take the exposure value as an example, call the service, and enter in the terminal,

```
rosservice call /camera/get_color_exposure
```

As shown in the figure below, the exposure value obtained is 100

```
jetson@yahboom:~$ rosservice call /camera/get_color_exposure
data: 100
success: True
message: ''
```

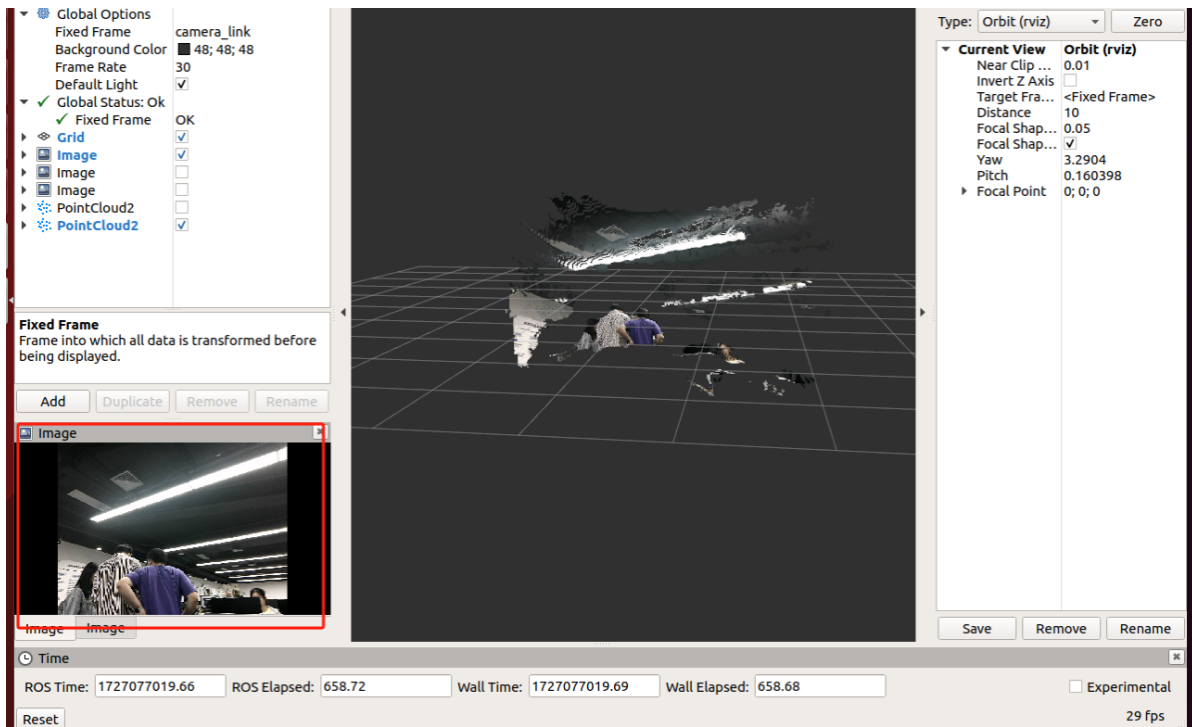
### 3.2.2, Camera parameter setting

Take setting the exposure value as an example, call the service, terminal input,

```
rosservice call /camera/set_color_exposure 50
```

```
jetson@yahboom:~$ rosservice call /camera/set_color_exposure 50
success: True [0.091, 0.09077219] point cloud unsubscribed
message: '' [0.091, 0.09077219] Image stream depth unsubscribed
jetson@yahboom:~$
```

After setting the exposure value to 50, the color image will change according to the set exposure value.



In the subsequent program development process, if the image we obtain is too bright or too dark, we can use commands to set the camera parameters to debug the image so that we can obtain the correct data.