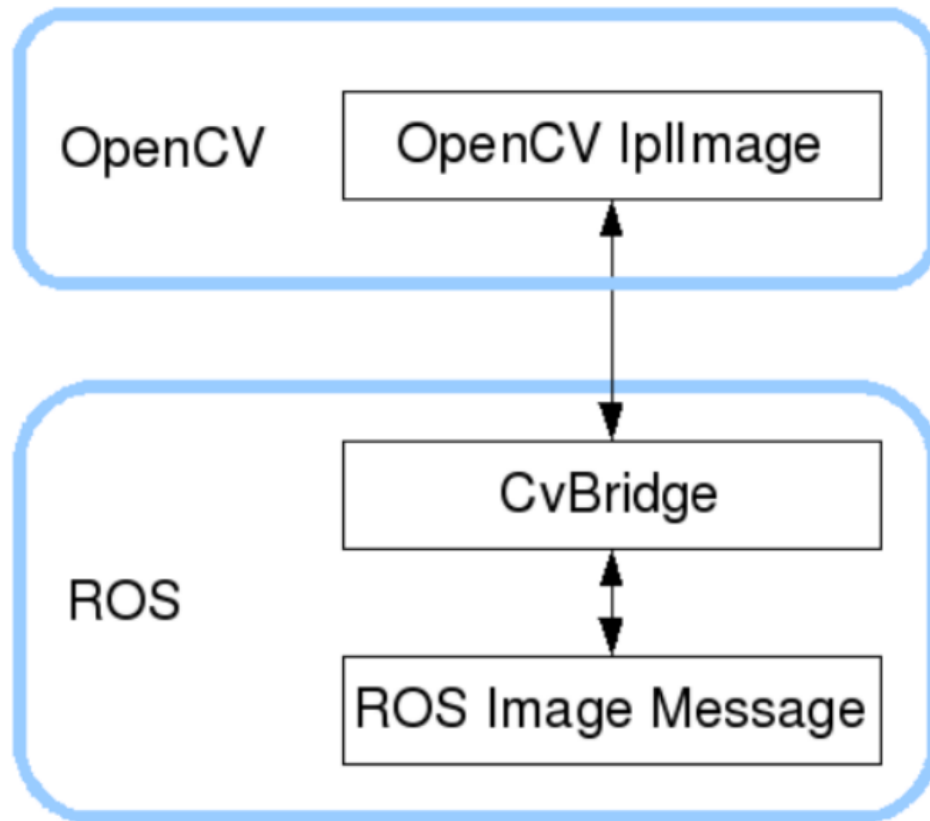


1.ROS+opencv application

ROS with its own sensor_ The msgs/Image message format transmits images and cannot be directly processed, but the provided CvBridge can perfectly convert and convert image data formats. CvBridge is an ROS library that serves as a bridge between ROS and Opencv.

The conversion of Opencv and ROS image data is shown in the following figure:



This lesson uses two case studies to demonstrate how to use CvBridge for data conversion.

1. Dabai_ DCW2 camera topic data

Input following command to start camera:

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

View the topic data list using the following command:

```
ros2 topic list
```

```
yahboom@VM:~/orbbec_ws$ 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
```

Among them, /camera/color/image_raw and /camera/depth/image_raw is the data of color and depth maps, and the data content of a certain frame can be viewed using the following command,

```
#View RGB image topic data content
ros2 topic echo /camera/color/image_raw
#Viewing Depth Image Topic Data Content
ros2 topic echo /camera/depth/image_raw
```

Color map:

```
header:
  stamp:
    sec: 1682406733
    nanosec: 552769817
    frame_id: camera_color_optical_frame
height: 480
width: 640
encoding: rgb8
is_bigendian: 0
step: 1920
data:
- 156
- 130
- 139
- 158
- 132
- 141
- 160
- 134
- 145
- 161
```

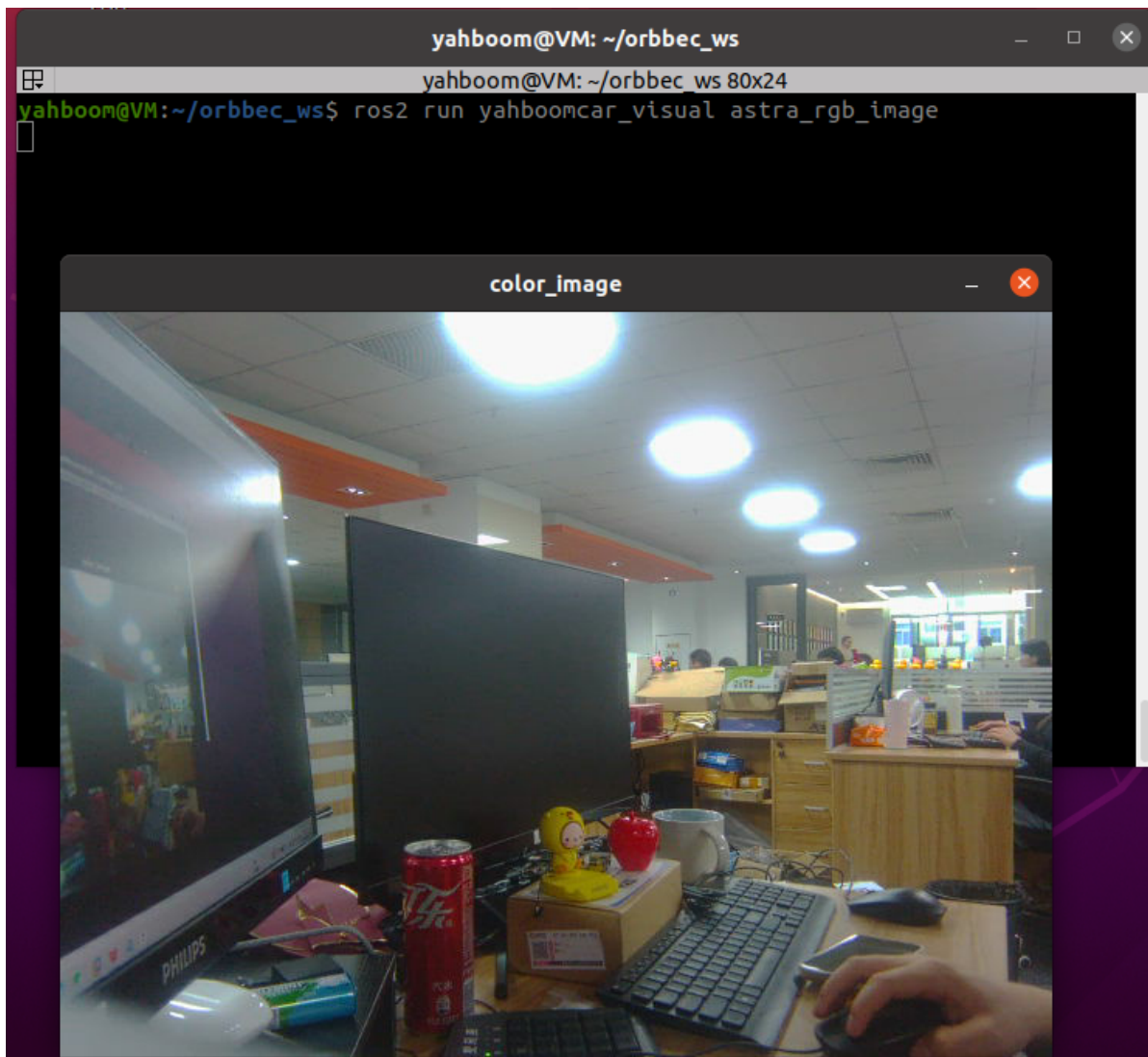
Depth map:

```
header:
  stamp:
    sec: 1682407553
    nanosec: 758139699
    frame_id: camera_depth_optical_frame
height: 480
width: 640
encoding: 16UC1
is_bigendian: 0
step: 1280
data:
- 0
- 0
- 0
- 0
- 226
- 17
- 226
- 17
```

2. Subscribe to color image topic data and display color images

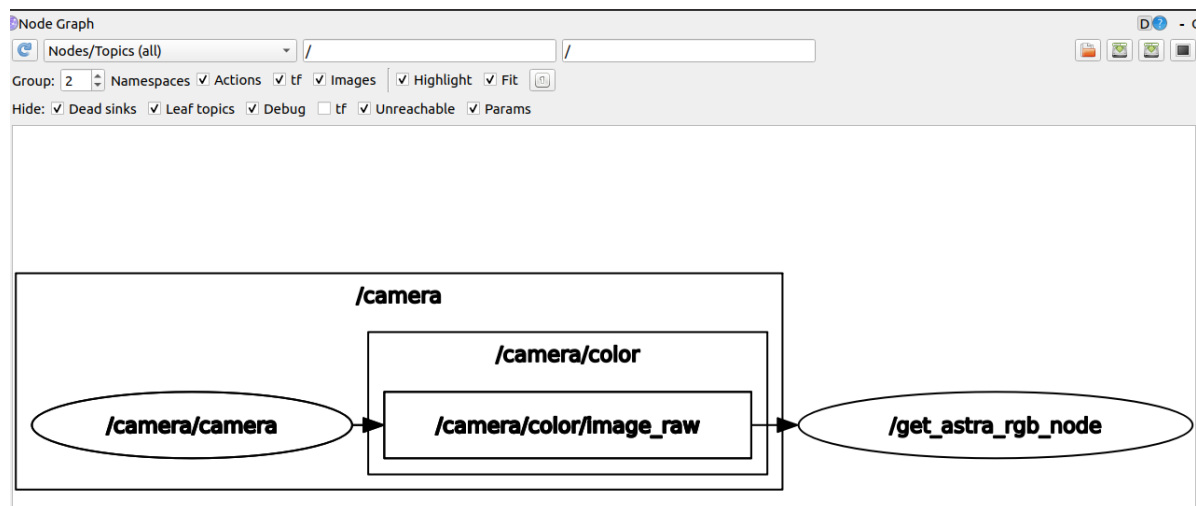
1) Input following command:

```
ros2 launch orbbec_camera dabai_dcw2.launch.py
ros2 run yahboomcar_visual astra_rgb_image
```



To view topic communication between nodes, input following command:

```
ros2 run rqt_graph rqt_graph
```



2) About code

Code path:

```
~/orbbec_ws/src/yahboomcar_visual/yahboomcar_visual/astra_rgb_image.py
```

/get_Astra_Rgb_Node node subscribed to/camera/color/image_Raw's topic, and then convert the topic data into car image data for publication through data conversion.

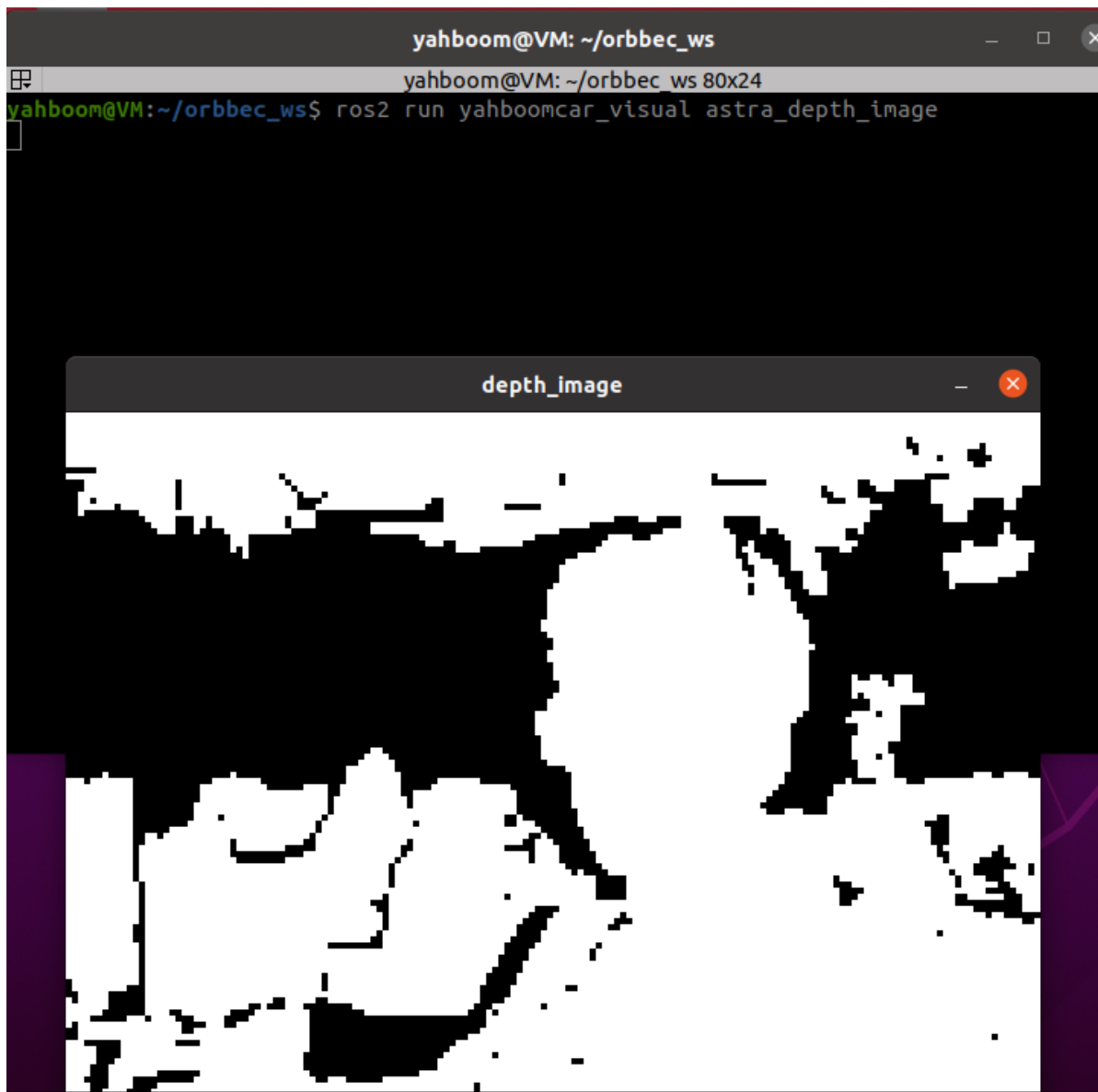
The code is as follows;

```
#Import opencv library and cv_Bridge Library
import cv2 as cv
from cv_bridge import CvBridge
#Creating CvBridge Objects
self.bridge = CvBridge()
#Define a subscriber to subscribe to RGB color image topic data published by deep
camera nodes
self.sub_img
=self.create_subscription(Image, '/camera/color/image_raw', self.handleTopic, 100)
#Convert msg to image data, where bgr8 is the image encoding format
frame = self.bridge.imgmsg_to_cv2(msg, "bgr8")
```

3. Subscribe to deep image topic information and display deep images

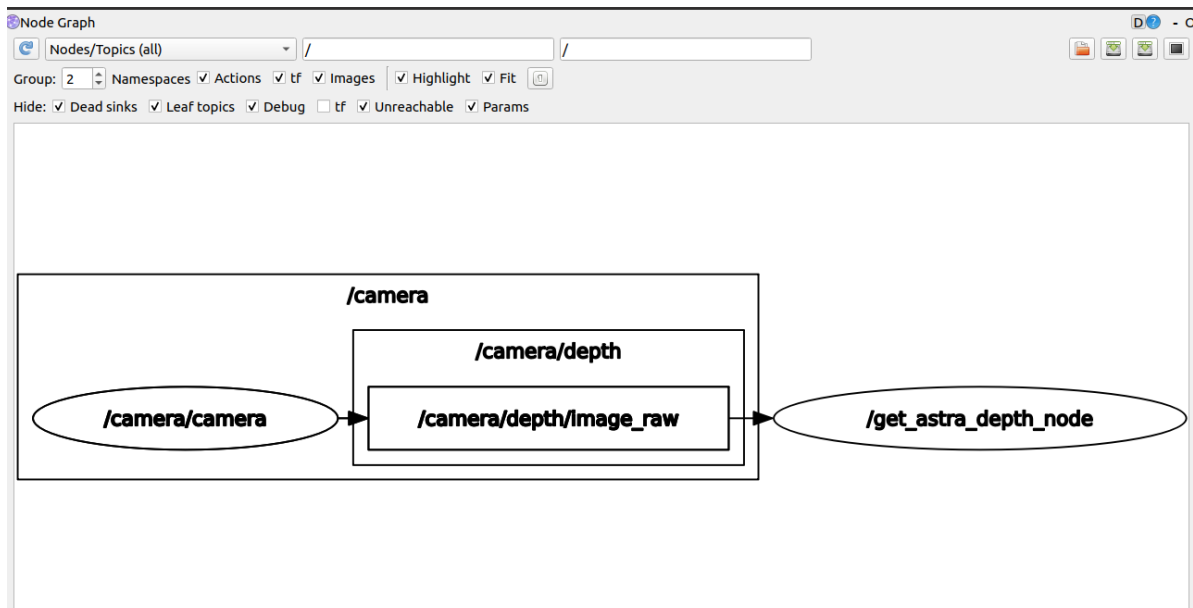
1) Input following command:

```
ros2 launch orbbec_camera dabai_dcw2.launch.py
ros2 run yahboomcar_visual astra_depth_image
```



To view topic communication between nodes, input following command:

```
ros2 run rqt_graph rqt_graph
```



2) About code

Code path:

```
~/orbbec_ws/src/yahboomcar_visual/yahboomcar_visual/astra_depth_image.py
```

The basic implementation process is the same as RGB color image display, subscribed to the/camera/depth/image published by the depth camera node_ Raw's topic data is then transformed into image data through data conversion.

The code is as follows;

```
#Import opencv library and cv_ Bridge Library
import cv2 as cv
from cv_bridge import CvBridge
#Creating CvBridge Objects
self.bridge = CvBridge()
#Define a subscriber to subscribe to RGB color image topic data published by deep
camera nodes
self.sub_img
=self.create_subscription(Image, '/camera/depth/image_raw', self.handleTopic, 10)
#Convert msg to image data, where bgr8 is the image encoding format
frame = self.bridge.imgmsg_to_cv2(msg, "32FC1")
```