2.QR code creation and recognition

1) Create QR code

- Code path:

```
~/orbbec_ws/src/yahboomcar_visual/simple_qrcode/QRcode_Create.py
```

- Input following command to got installation package:

```
python3 -m pip install qrcode pyzbar
sudo apt-get install libzbar-dev
```

- Start

```
cd ~/orbbec_ws/src/yahboomcar_visual/simple_qrcode
python QRcode_Create.py
```

After the program runs, it will prompt for the generated content, and press the Enter key to confirm the content.

Here, taking the creation of the "yahboom" string as an example.

- Code

```
#Create qrcode object
qr = qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_H,
    box_size=5,
    border=4,)
#各参数含义  Meaning of each parameter
'''version：值为1~40的整数，控制二维码的大小（最小值是1，是个12×12的矩阵）。
如果想让程序自动确定，将值设置为 None 并使用 fit 参数即可。
error_correction：控制二维码的错误纠正功能。可取值下列4个常量。
ERROR_CORRECT_L：大约7%或更少的错误能被纠正。
ERROR_CORRECT_M（默认）：大约15%或更少的错误能被纠正。
ROR_CORRECT_H：大约30%或更少的错误能被纠正。
box_size：控制二维码中每个小格子包含的像素数。
border：控制边框（二维码与图片边界的距离）包含的格子数（默认为4，是相关标准规定的最小值）
Version: An integer with a value of 1-40 that controls the size of the QR
code (the minimum value is 1, which is a 12 × 12 matrix).
If you want the program to automatically determine, set the value to None
and use the fit parameter.
Error_ Correction: Control the error correction function of the QR code. The
following four constants can be taken as values.
ERROR_ CORRECT_ L: Approximately 7% or less of errors can be corrected.
ERROR_ CORRECT_ M (default): Approximately 15% or less of errors can be
corrected.
ROR_ CORRECT_ H: Approximately 30% or less of errors can be corrected.
Box_ Size: Controls the number of pixels contained in each small grid in the
QR code.
```

```
Border: Controls the number of cells contained in the border (the distance
between the QR code and the image boundary) (default is 4, which is the
minimum value specified by relevant standards)'''
#qrcode二维码添加logo  Adding logo to qrcode QR code
my_file = Path(logo_path)
if my_file.is_file(): img = add_logo(img, logo_path)
#添加数据  Add data
qr.add_data(data)
# 填充数据
# fill data
qr.make(fit=True)
# 生成图片
# generate images
img = qr.make_image(fill_color="green", back_color="white")
```

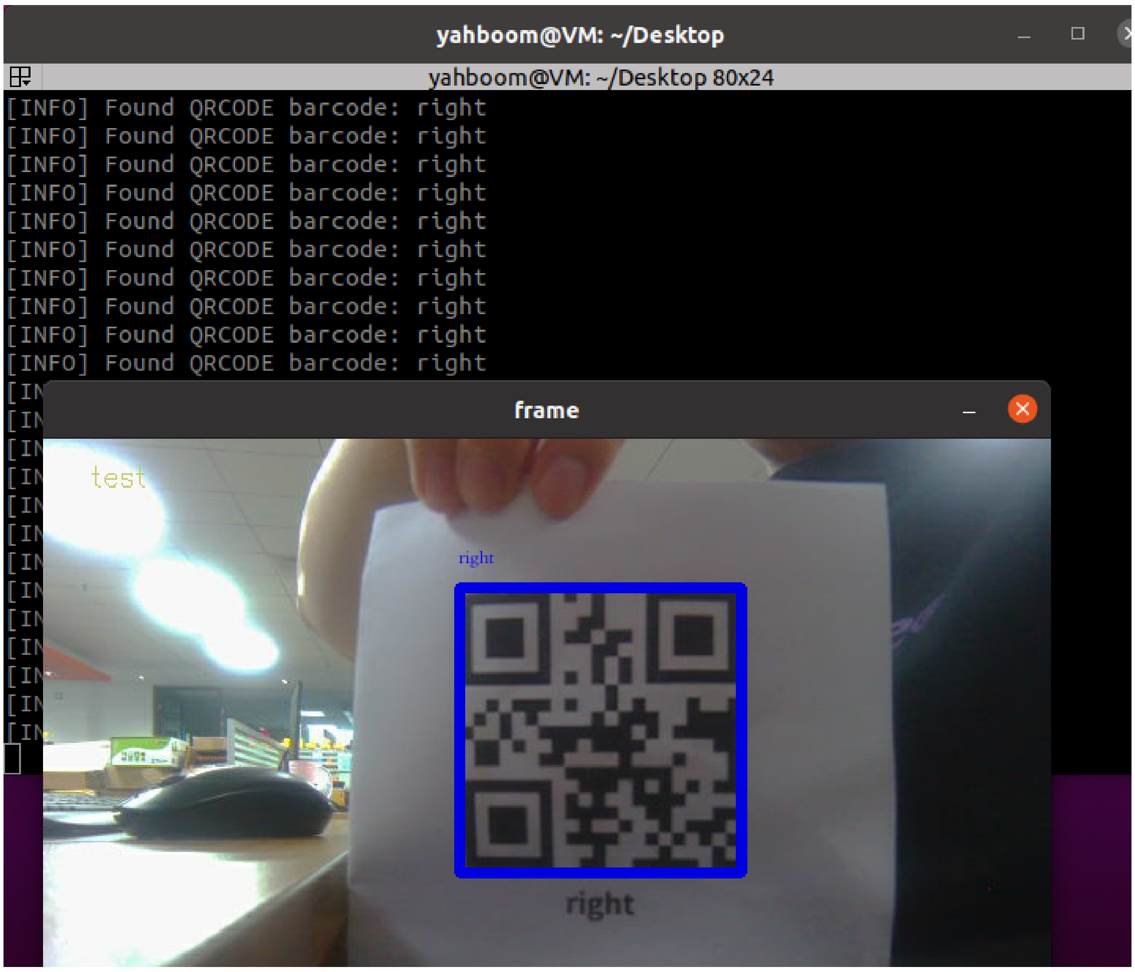
2) Identify QR code

- Code path:

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

- Start

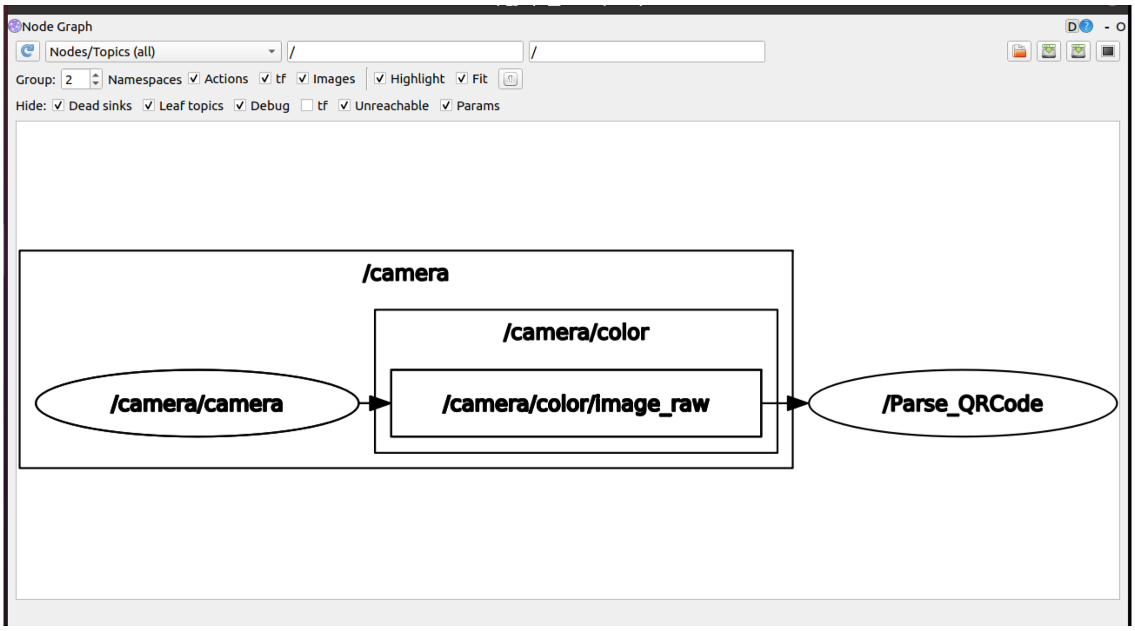
```
ros2 launch orbbec_camera dabai_dcw2.launch.py
ros2 run yahboomcar_visual QRcode_Parsing
```



```
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
[INFO] Found QRCODE barcode: right
```

- View node communication diagram

```
ros2 run rqt_graph rqt_graph
```



- Code

```
#订阅彩色图像信息
#Subscribe to color image information
self.sub_img =
self.create_subscription(CamImage,'/camera/color/image_raw',self.handleTopic
,100)
#在回调函数中把图像数据传递到解析图像函数中
#Transfer image data to the parsing image function in the callback function
frame = self.decodeDisplay(frame)
#解析图像
#Analyzing images
def decodeDisplay(self,image):
    gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
    # 需要先把输出的中文字符转换成Unicode编码形式
    # The output Chinese characters need to be converted to Unicode encoding
first
    barcodes = pyzbar.decode(gray)
    for barcode in barcodes:
        # 提取二维码的边界框的位置
        # Extract the position of the boundary box of the TWO-DIMENSIONAL
code
        # 画出图像中条形码的边界框
        # Draw the bounding box for the bar code in the image
        (x, y, w, h) = barcode.rect
        cv.rectangle(image, (x, y), (x + w, y + h), (225, 0, 0), 5)
        encoding = 'UTF-8'
        # 画出来，就需要先将它转换成字符串
        # to draw it, you need to convert it to a string
        barcodeData = barcode.data.decode(encoding)
        barcodeType = barcode.type
        # 绘出图像上数据和类型
        # Draw the data and type on the image
        pilimg = Image.fromarray(image)
        # 创建画笔
        # create brush
        draw = ImageDraw.Draw(pilimg)  # 图片上打印  Print on picture
        # 参数1：字体文件路径，参数2：字体大小
        # parameter 1: font file path, parameter 2: font size
        fontStyle =
ImageFont.truetype("/home/yahboom/orbbec_ws/src/yahboomcar_visual/yahboomcar
_visual/Block_Simplified.TTF", size=12, encoding=encoding)
        # # 参数1：打印坐标，参数2：文本，参数3：字体颜色，参数4：字体
        # Parameter 1: print coordinates, parameter 2: text, parameter 3:
font color, parameter 4: font
        draw.text((x, y - 25), str(barcode.data, encoding), fill=(255, 0,
0), font=fontStyle)
        # # PIL图片转cv2 图片
        # PIL picture to CV2 picture
        image = np.array(pilimg)
        # 向终端打印条形码数据和条形码类型
        # Print barcode data and barcode type to terminal
        print("[INFO] Found {} barcode: {}".format(barcodeType,
barcodeData))
        return image
```