4. AR vision

4.1. AR Overview

Augmented Reality (AR), referred to as "AR", is a technology that cleverly integrates virtual information with the real world, widely using multimedia, three-dimensional modeling, real-time tracking and registration, intelligent interaction, sensing and other technical means. The computer generated text, image, 3D model, music, video and other virtual information after simulation, applied to the real world, the two kinds of information complement each other, so as to achieve the real world "enhancement".

AR system has three outstanding characteristics: (1) Information integration between real world and virtual world; ② It has real-time interaction; ③ is to add positioning virtual objects in the three-dimensional scale space.

Augmented reality technology includes multimedia, three-dimensional modeling, real-time video display and control, multi-sensor fusion, real-time tracking and registration, scene fusion and other new technologies and new means.

4.2 Program function description

Once the program is started, the Astra camera recognizes the checkerboard and displays the AR effect, which can be visualized on the web side.

The operation of this function must have the internal parameters of the camera, the internal parameters file path is,

```
/userdata/yahboomcar_ws/src/yahboomcar_visual/astra.yaml
```

The internal reference calibration can be quickly calibrated with checkerboard, and the specific method can be seen in the content of the "Camera internal reference calibration" in the "Depth Camera Series Course". **(This step has already been done in the system image)**

After the calibration is completed on the VM, move the generated [calibrationdata.tar.gz] file to the current directory to decompress it, open the [ost.yaml] file in the folder, and find the camera parameter matrix and distortion coefficient.

After remotely connecting the car in VSCode, open the [/userdata/yahboomcar_ws/src/yahboomcar_visual/astra.yaml] file and modify the following two [data] contents (if you are familiar with linux instructions, You can also directly switch to the file directory after SSH connection to the car in the terminal, and use the vim tool to modify the file content),

```
camera_matrix: !!opencv-matrix
  rows: 3
  cols: 3
  dt: d
  data: [608.182089,   0.      , 324.255471,
           0.      , 607.855617, 245.971079,
           0.      ,   0.      ,   1.      ]
distortion_model: plumb_bob
distortion_coefficients: !!opencv-matrix
  rows: 1
  cols: 5
  dt: d
  data: [0.100236, -0.073376, 0.004033, -0.001132, 0.000000]
```

There are a total of 12 effects in this section.

```
["Triangle", "Rectangle", "Parallelogram", "WindMill", "TableTennisTable",
"Ball", "Arrow", "Knife", "Desk", "Bench", "Stickman", "ParallelBars"]
```

4.3 Program Code Reference paths

After SSH connection car, the location of the function source code is located at,

```
#launch文件#launch file
/userdata/yahboomcar_ws/src/yahboomcar_visual/launch/ar_launch.xml
```

The node or launch file is explained as follows:

- simple_AR: Generate AR effect. The source path is,

```
/userdata/yahboomcar_ws/src/yahboomcar_visual/yahboomcar_visual/simple_AR.py
```

- rosbridge_websocket_launch.xml: enables the websocket to interact with the Web
- rosboard_node: starts the ROSboard

The program uses opencv to read camera image data, so it does not start the ROS node of Astra's camera.

4.4. Program Startup

After SSH connection to the car, terminal input,,

```
ros2 launch yahboomcar_visual ar_launch.xml
```

View ros topics by using the following command, enter in the virtual machine terminal,

```
ros2 topic list
```

```
yahboom@yahboom-virtual-machine:~$ ros2 topic list
/Graphics_topic
/client_count
/connected_clients
/key_value
/mouse_pos
/parameter_events
/rosout
/simpleAR/camera
```

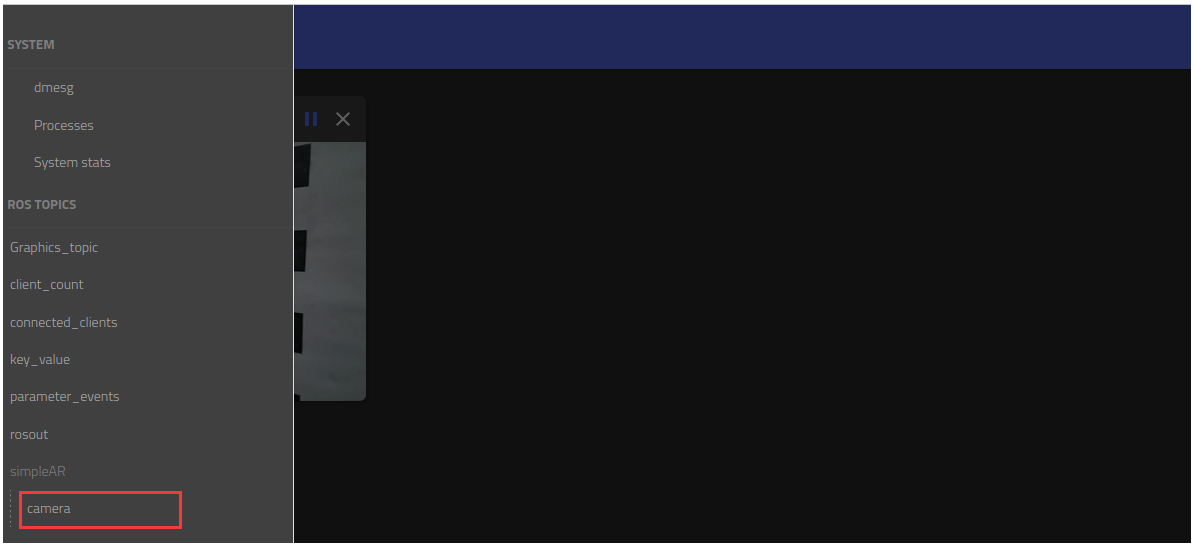
- /Graphics_topic: The topic name of the effect. Subscribe to the effect to be identified.
- /simpleAR/camera: The topic name of the image to publish the image.

4.4.1. Web display

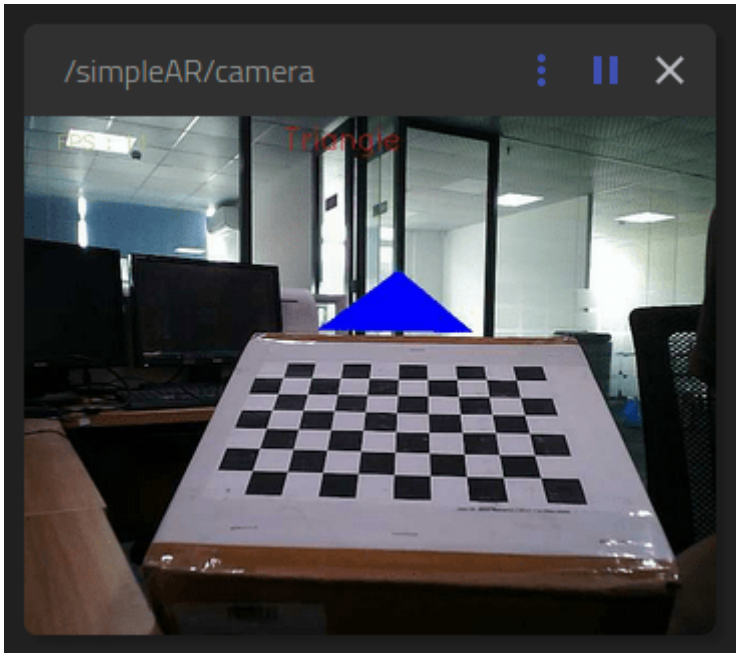
Open the browser on the PC side (note that the computer and the Rising Sun network must be in the same LAN), enter the URL: car IP:9999, for example, my car IP is 192.168.2.67, enter the URL in the browser on the virtual machine side to open the ROSboard webpage:

```
192.168.2.67:9999
```

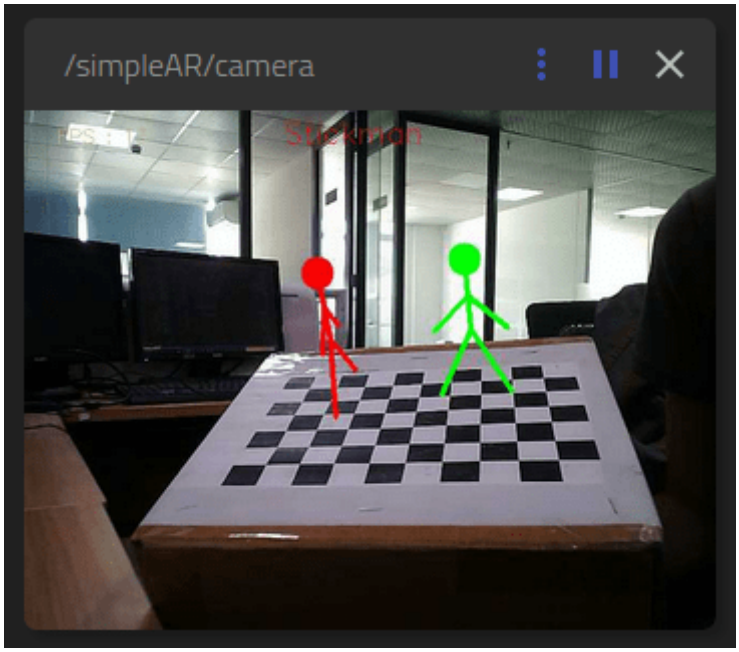
Select the topic /simpleAR/camera,



You can show the AR effect.



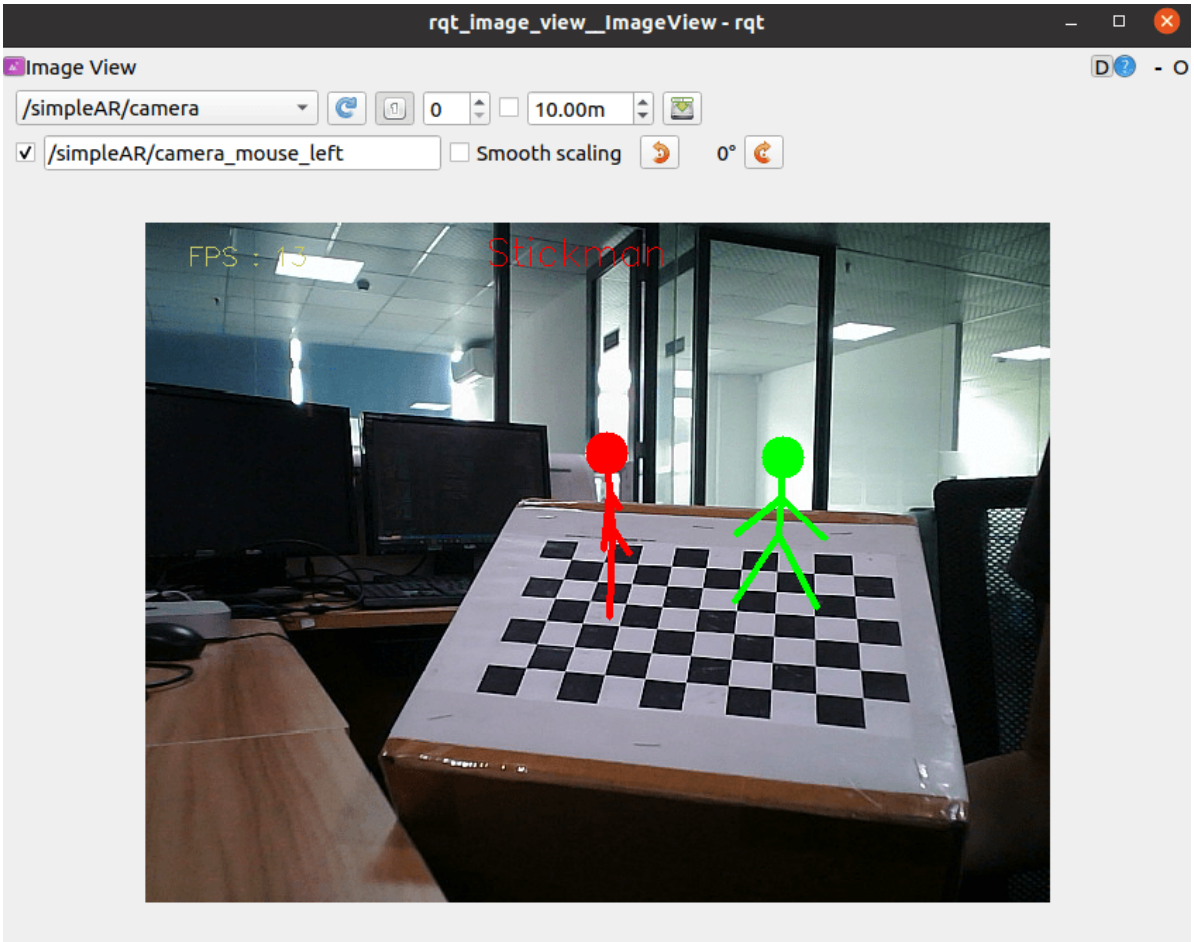
Press [f] key in the page to switch different effects.



4.4.2 rqt_image_view

You can also use rqt_image_view to view published images, virtual machine terminal input,

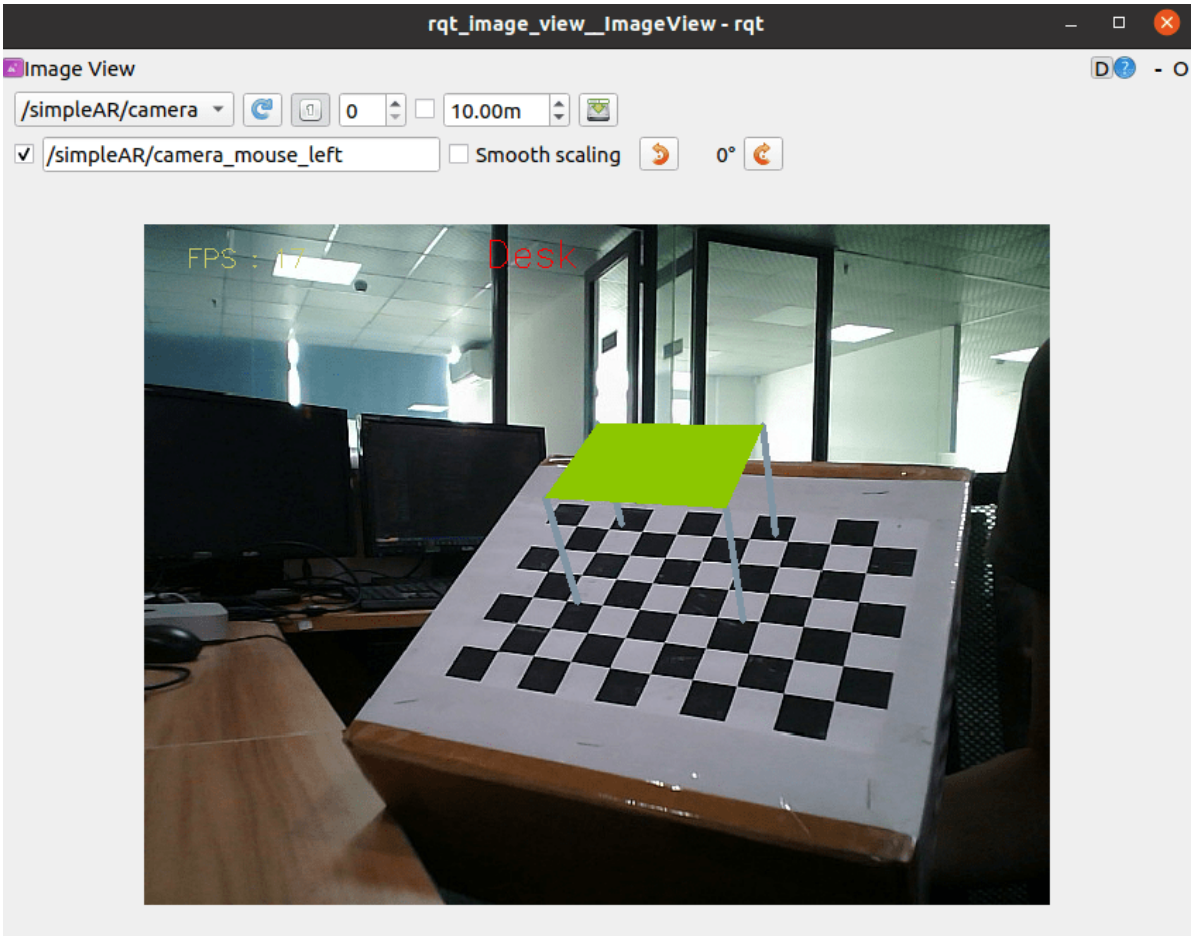
```
ros2 run rqt_image_view rqt_image_view
```



In the top left corner, select /simpleAR/camera to view the image.

At this time, the effect cannot be switched by keyboard keys, but it can be modified by publishing topic data, for example, virtual machine terminal input,

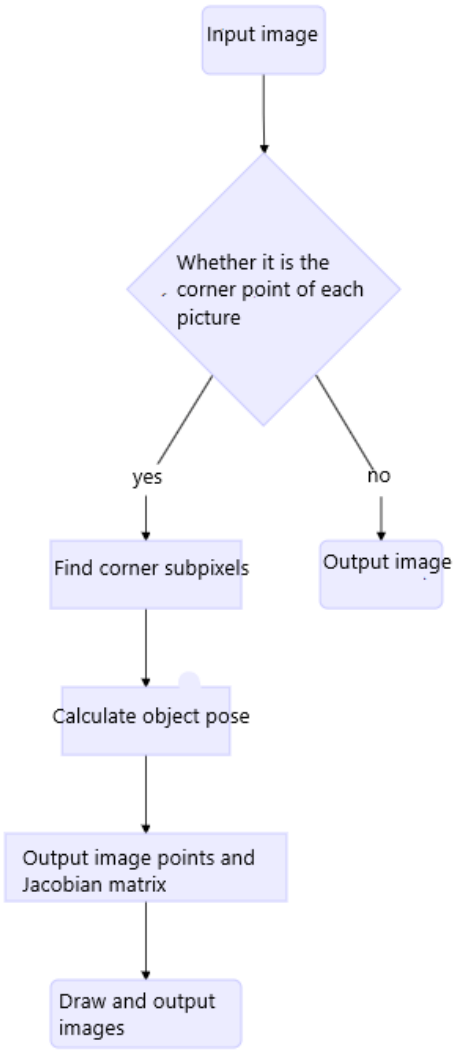
```
ros2 topic pub /Graphics_topic std_msgs/msg/String "data: Desk"
```



At this point the AR effect was changed to Desk.

4.5. Core Source Code Resolution

Code flow:



* Part of the code in simple_AR.py is as follows:

```
# 查找每个图片的角点
# Find the corner of each image
retval, corners = cv.findChessboardCorners(
    gray, self.patternSize, None,
    flags=cv.CALIB_CB_ADAPTIVE_THRESH + cv.CALIB_CB_NORMALIZE_IMAGE +
cv.CALIB_CB_FAST_CHECK)
# 查找角点亚像素
# Find corner subpixels
if retval:
    corners = cv.cornerSubPix(
        gray, corners, (11, 11), (-1, -1),
        (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 30, 0.001))
    # 计算对象姿态
    # Compute object pose
    retval, rvec, tvec, inliers = cv.solvePnPRansac(
        self.objectPoints, corners, self.cameraMatrix, self.distCoeffs)
    # 输出图像点和雅可比矩阵
    # Output image points and Jacobian matrix
    image_Points, jacobian = cv.projectPoints(
        self.axis, rvec, tvec, self.cameraMatrix, self.distCoeffs, )
    img = self.draw(img, corners, image_Points)
# 发布图像
# Publish image
self.pub_img.publish(self.bridge.cv2_to_imgmsg(img, "bgr8"))
```