AR Vision

1 Overview

Augmented Reality, referred to as "AR", is a technology that cleverly integrates virtual information with the real world. It widely uses multimedia, three-dimensional modeling, real-time tracking and registration, intelligent interaction, sensing and other technologies. It simulates computer-generated text, images, three-dimensional models, music, videos and other virtual information and then applies it to the real world. The two types of information complement each other, thereby achieving "enhancement" of the real world.

The AR system has three outstanding characteristics: ① information integration between the real world and the virtual world; ② real-time interactivity; ③ adding positioning virtual objects in the three-dimensional scale space.

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

2. How to use

It is best to log in to the Mechanical Dog desktop through vnc for operation. Mechanical Dog vnc is used. Reference: 14. Radar Mapping Navigation\6. ROS2 Environment Entity Mechanical Dog Status Acquisition\ROS2 Environment Acquisition of Mechanical Dog Real Joint Data.pdf

When using AR cases, you must have the internal parameters of the camera, otherwise it will not work. The internal parameter files are in the same directory as the code.

After the calibration work is completed, a [calibrationdata.tar.gz] will be generated, and the [calibrationdata.tar.gz] file will be moved to the [home] directory. After decompression, open [ost.yaml] in the folder, find the camera internal parameter matrix and distortion coefficient and modify them to the corresponding locations of the [astra.yaml] file. You only need to modify the contents of two [data]. For example: the following,

```
camera_matrix: !!opencv-matrix
rows: 3
cols: 3
dt: d
data: [615.50506, 0. , 365.84388,
0. , 623.69024, 238.778 ,
0. , 0. , 1. ]
distortion_model: plumb_bob
distortion_coefficients: !!opencv-matrix
rows: 1
cols: 5
dt: d
data: [0.166417, -0.160106, -0.008776, 0.025459, 0.000000]
```

There are a total of 12 effects in the cases in this section.

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

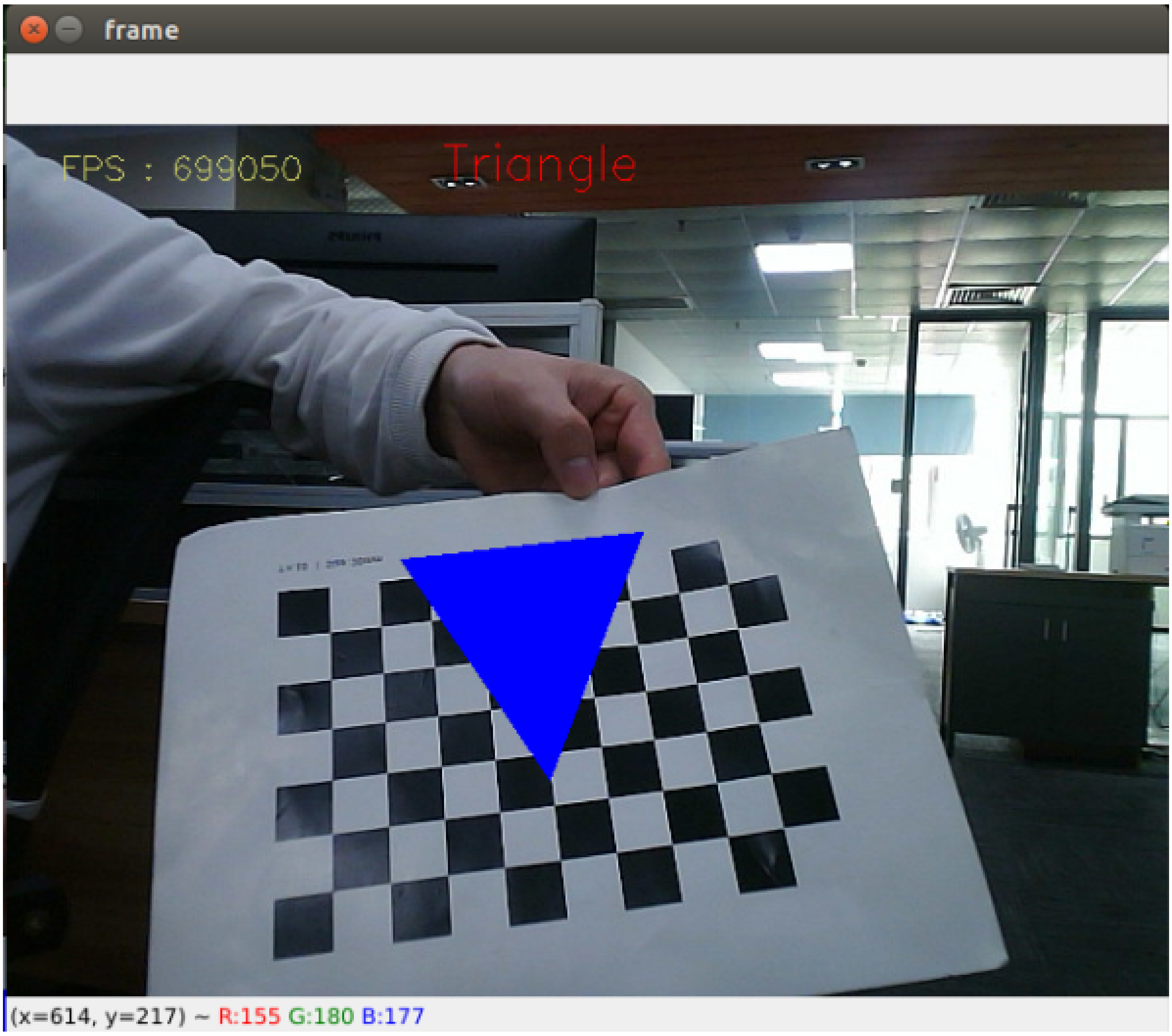
3. Start command

Code reference path,

```
#pi4
/home/pi/cartographer_ws2/src/yahboom_visual/yahboom_visual/simple_AR.py
#pi5
/root/yahboomcar_ws/src/yahboom_visual/yahboom_visual/simple_AR.py
```

Open the robot dog terminal and enter,

```
#pi4
cd /home/pi/cartographer_ws2/
source install/setup.bash
ros2 run yahboom_visual simple_AR
#pi5 (need to enter the same docker terminal)
ros2 run yahboom_visual simple_AR
```



(x=614, y=217) ~ R:155 G:180 B:177

[q] key to exit, [f] key to switch different effects.

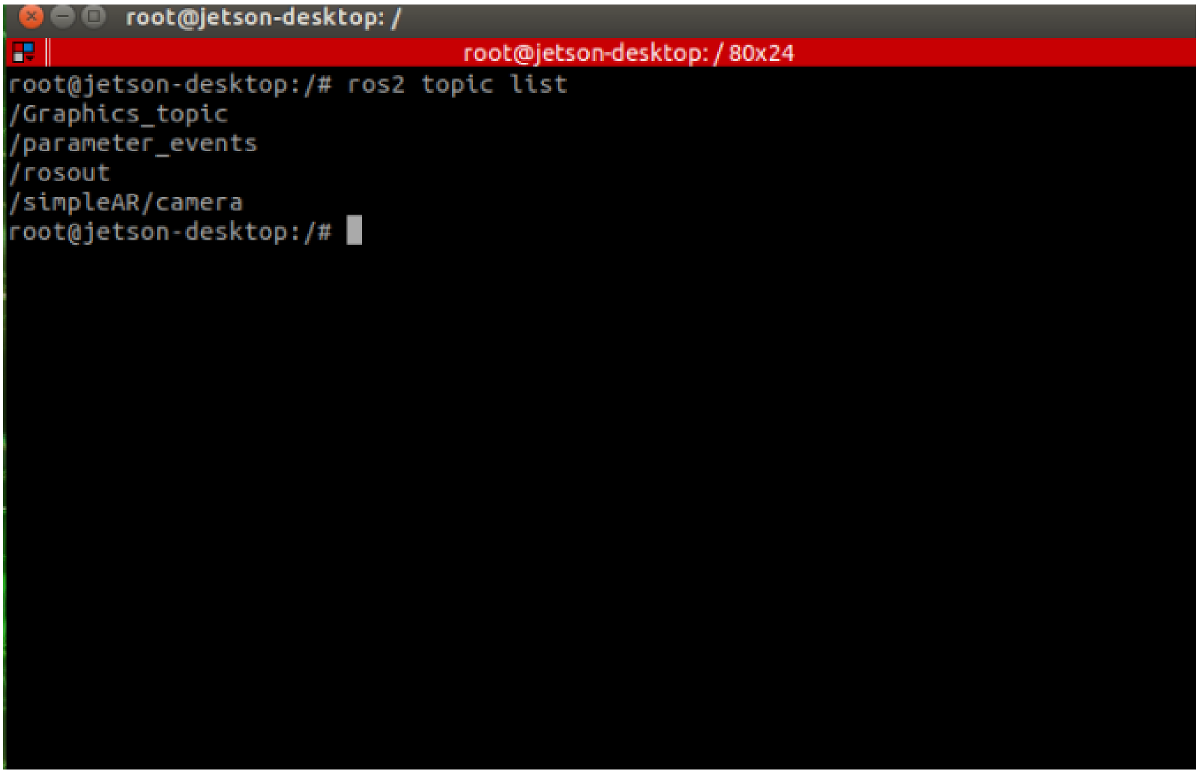
3.1.1, ROS deployment

This course also deploys ROS, which mainly has the following two functions:

- Subscribe to topic data and switch between different effects
- Post images

View the ros topic through the following command, input in the mechanical dog terminal,

```
#pi4
ros2 topic list
#pi5 (need to enter the same docker terminal)
ros2 topic list
```

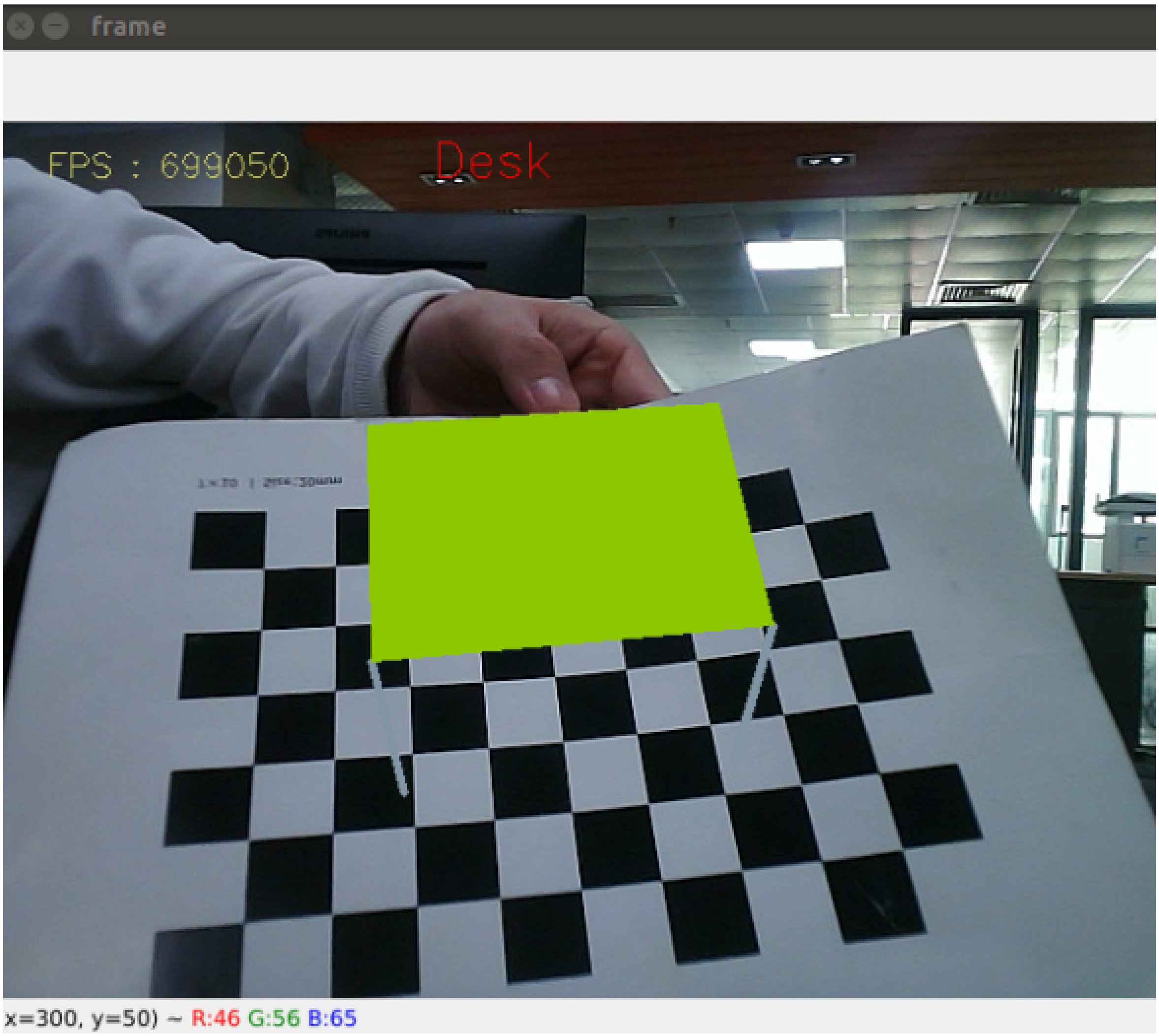


```
root@jetson-desktop:/# ros2 topic list
/Graphics_topic
/parameter_events
/rosout
/simpleAR/camera
root@jetson-desktop:/# 
```

- /Graphics_topic: The topic name of the effect, the effect that needs to be identified when subscribing.
- /simpleAR/camera: The topic name of the image, publish the image.

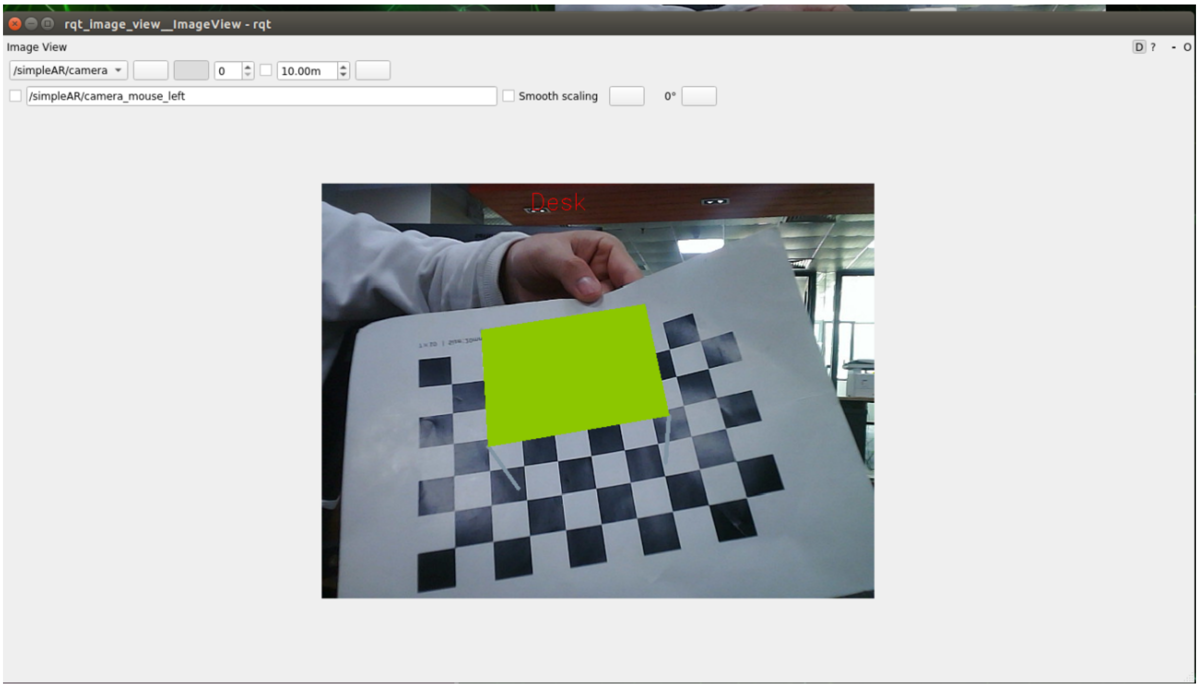
The modification effect can be modified through the following command. For example, I first changed it to Desk and entered it in the terminal.

```
#pi4
ros2 topic pub /Graphics_topic std_msgs/msg/String "data: Desk"
#pi5 (need to enter the same docker terminal)
ros2 topic pub /Graphics_topic std_msgs/msg/String "data: Desk"
```



To view the published image, you can use rqt_image_view to view it and input it into the mechanical dog terminal.

```
#pi4
ros2 run rqt_image_view rqt_image_view
#pi5 (need to enter the same docker terminal)
ros2 run rqt_image_view rqt_image_view
```



Select the topic /simpleAR/camera in the upper left corner to view the image.