5. AR Vision

1. Overview

Augmented Reality, referred to as "AR", technology is a kind of virtual information and the real world clever fusion of technology, a wide range of multimedia, three-dimensional modelling, real-time tracking and registration, intelligent interaction, sensing, and other technological means, computer-generated text, images, three-dimensional models, music, video and other virtual information simulation simulation applied to the real world, the two kinds of information complement each other, so as to achieve the "augmentation" of the real world. The computer-generated text, images, three-dimensional models, music, video and other virtual information simulation 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 features: ① real world and virtual world information integration; ② real-time interactivity; ③ in the three-dimensional scale space to add positioning virtual objects.

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

2. Usage

When using the AR case, you must have the internal reference of the camera, otherwise it can not run. The internal reference file is in the same directory as the code.

After the calibration work is completed, a [calibrationdata.tar.gz] will be generated, move the [calibrationdata.tar.gz] file to the [home] directory. After unzipping, open the [ost.yaml] inside the folder, find the camera internal reference matrix and distortion coefficients and modify them to the corresponding positions in the [astra.yaml] file, only need to modify the contents of the two [data]. For example, the following content.

```
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 this section of the case.

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

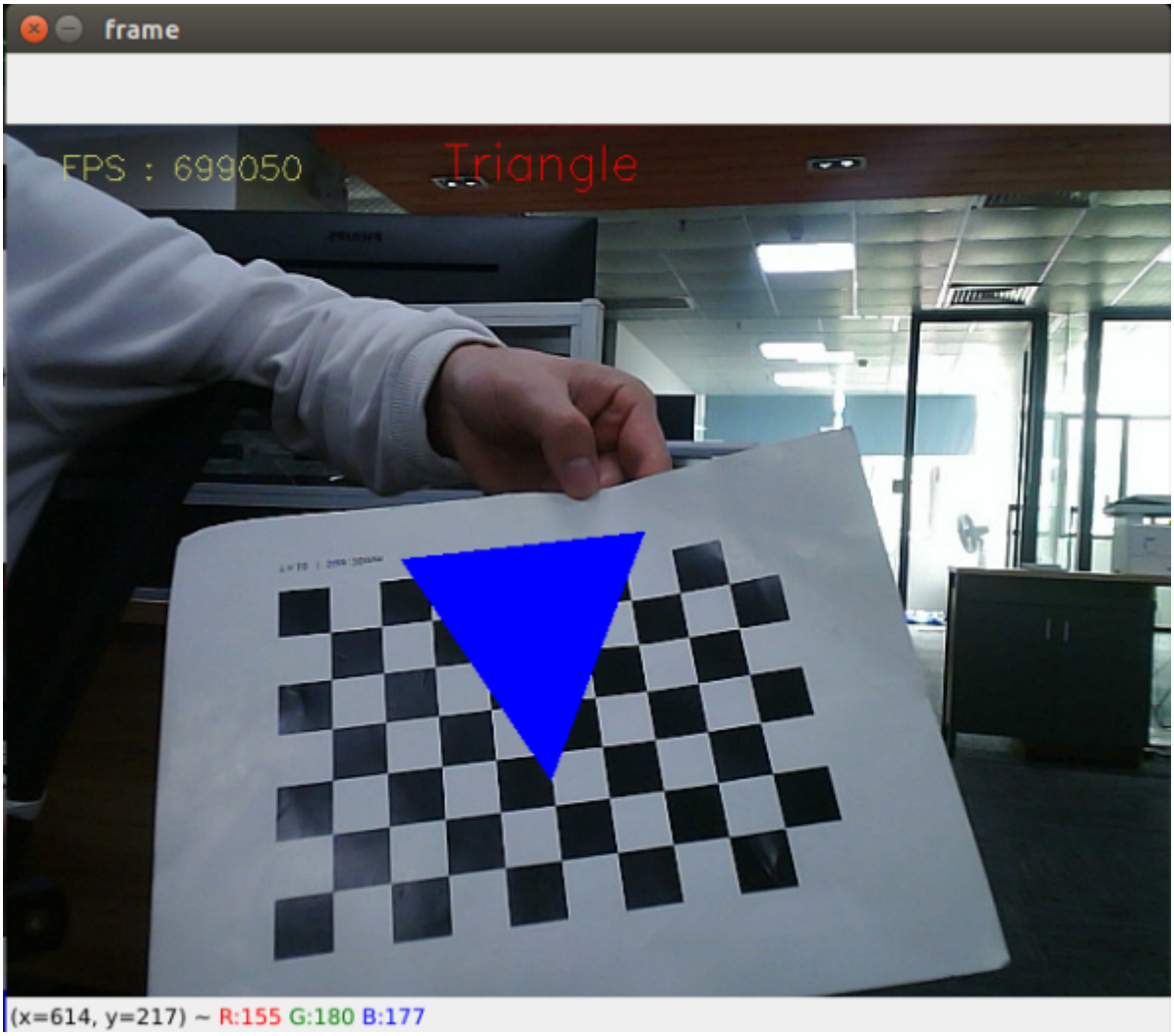
3. Startup commands

The code reference path.

```
/home/pi/cartographer_ws2/src/yahboom_visual/yahboom_visual/simple_AR.py
```

After entering the docker container, docker terminal type, the

```
ros2 run yahboom_visual simple_AR
```



The [q] key exits and the [f] key switches between different effects.

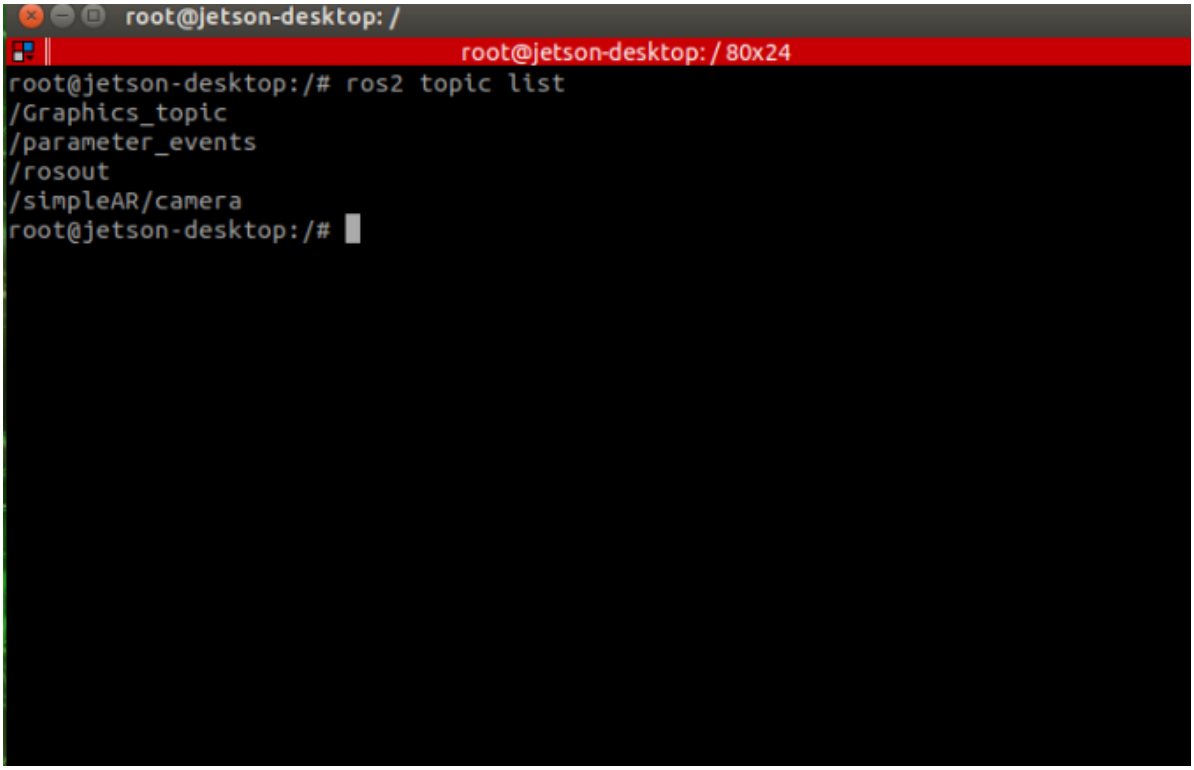
3.1.1 ROS Deployment

This section of the course also deploys ROS, mainly with the following two functions:

- Subscribing to topic data, switching different effects
- Publish images

View ros topics by the following command, docker terminal input.

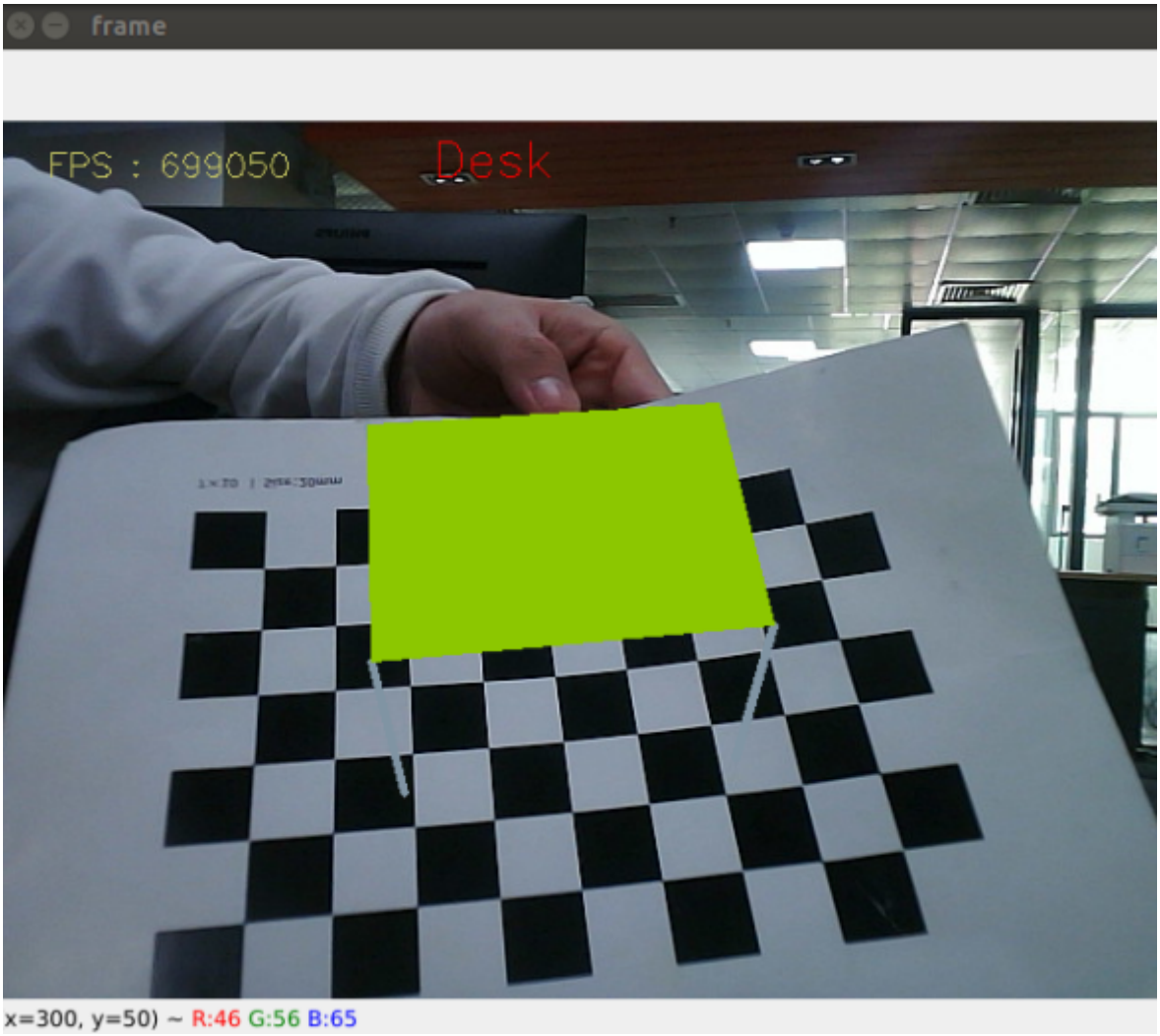
```
ros2 topic list
```



- /Graphics_topic: topic name of the effect, subscribe to the effect to be recognised.
- /simpleAR/camera: topic name of the image, publish the image.

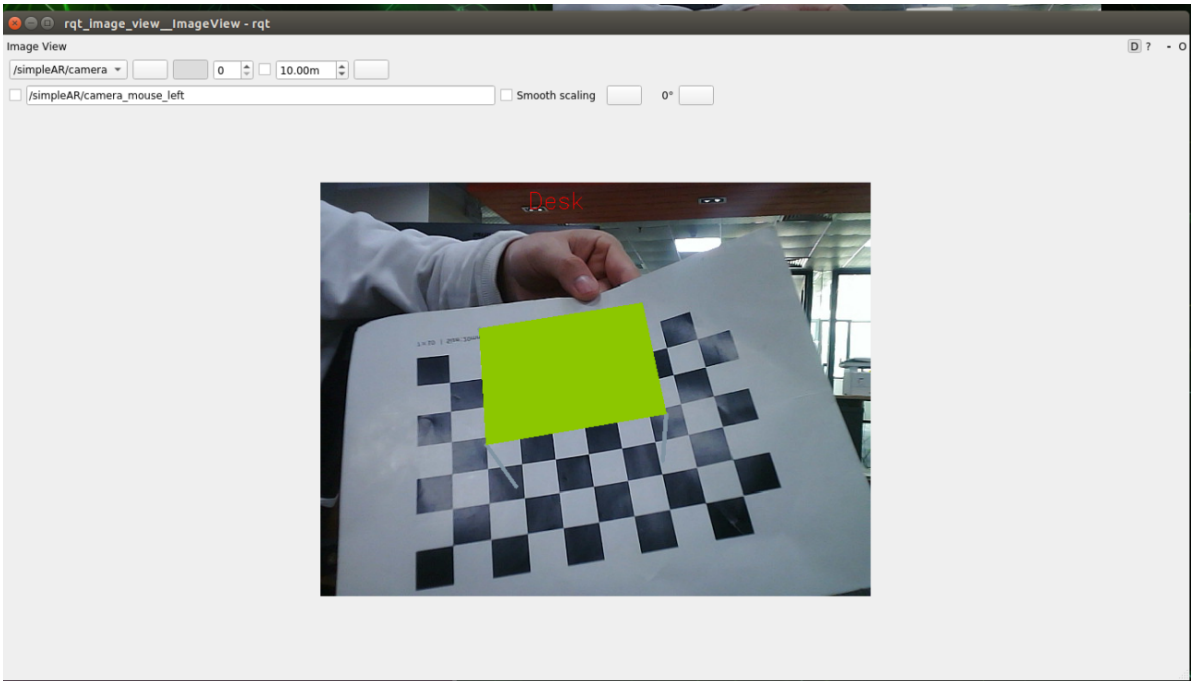
To modify the effect, you can modify it with the following commands, for example, I'll start by modifying it to Desk, docker terminal input.

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



Viewing the published image can be done using rqt_image_view, docker terminal input, the

```
ros2 run rqt_image_view rqt_image_view
```



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