# AR vision

## 1、Overview

Augmented Reality, referred to as "AR", is a technology that ingeniously integrates virtual information with the real world. It widely uses multimedia, 3D modeling, real-time tracking and registration, intelligent interaction, sensing and other technologies. The method is to simulate and simulate virtual information such as text, images, three-dimensional models, music, and videos generated by the computer, and then apply it to the real world. The two kinds of information complement each other, thereby realizing the "enhancement" of the real world.

The AR system has three prominent features: (1) the information integration of the real world and the virtual world; (2) real-time interactivity; (3) the addition of 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.

## 2、How to use

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, and different cameras correspond to different internal parameters.Internal parameter calibration can be quickly calibrated using a checkerboard. The specific method can be found in the first section [Monocular Camera Calibration] in [7. Monocular Camera Series Courses]. **(This step has already been completed in the docker image)**

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:
  rows: 3
  cols: 3
  data: [524.29384,   0.     , 543.91968,
           0.     , 557.31848, 273.30891,
           0.     ,   0.     ,   1.     ]
distortion_model: plumb_bob
distortion_coefficients:
  rows: 1
  cols: 5
  data: [-0.406832, 0.080272, -0.002452, -0.062491, 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"]
```

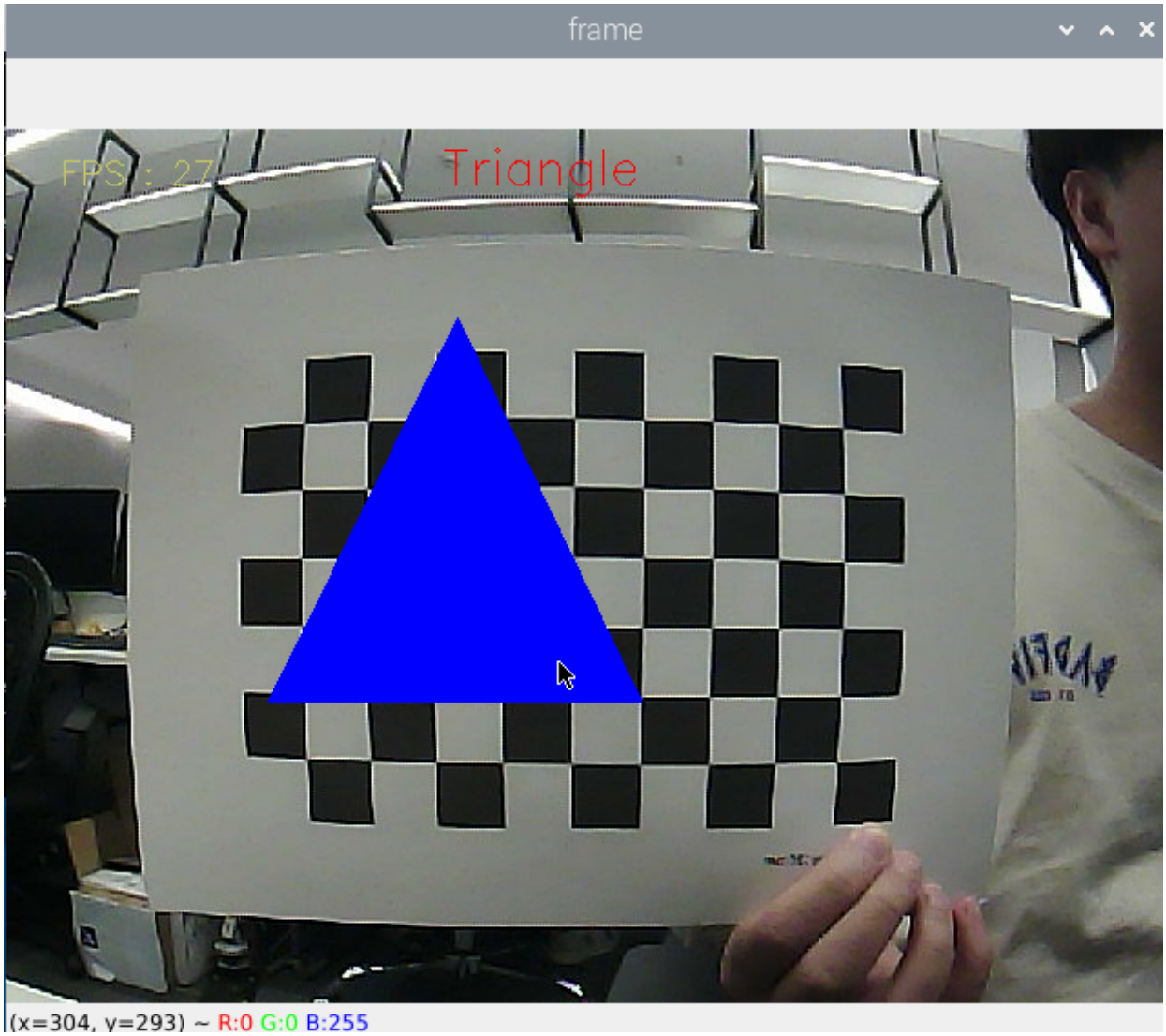
## 3、Run command

Code reference path

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

After entering the docker container, enter the following command in the docker terminal

```
ros2 run yahboomcar_visual simple_AR
```




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

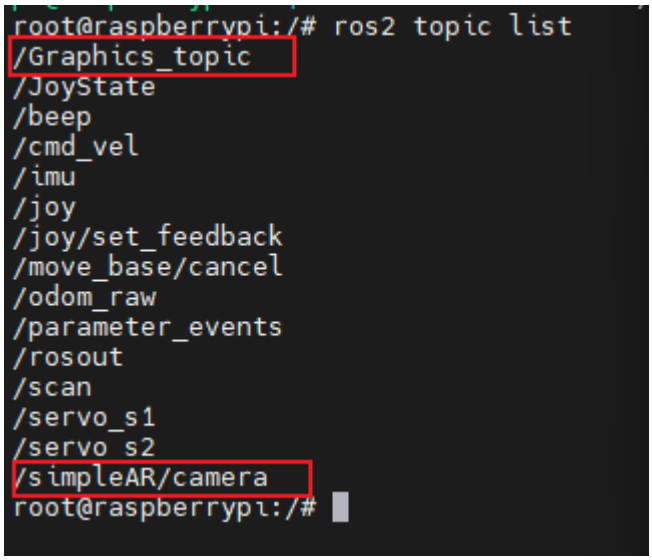
### 3.1.1、ROS deployment

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

- Subscribe to topic data and switch between different effects
- post image

View the ros topic through the following command, enter the following command in the docker terminal

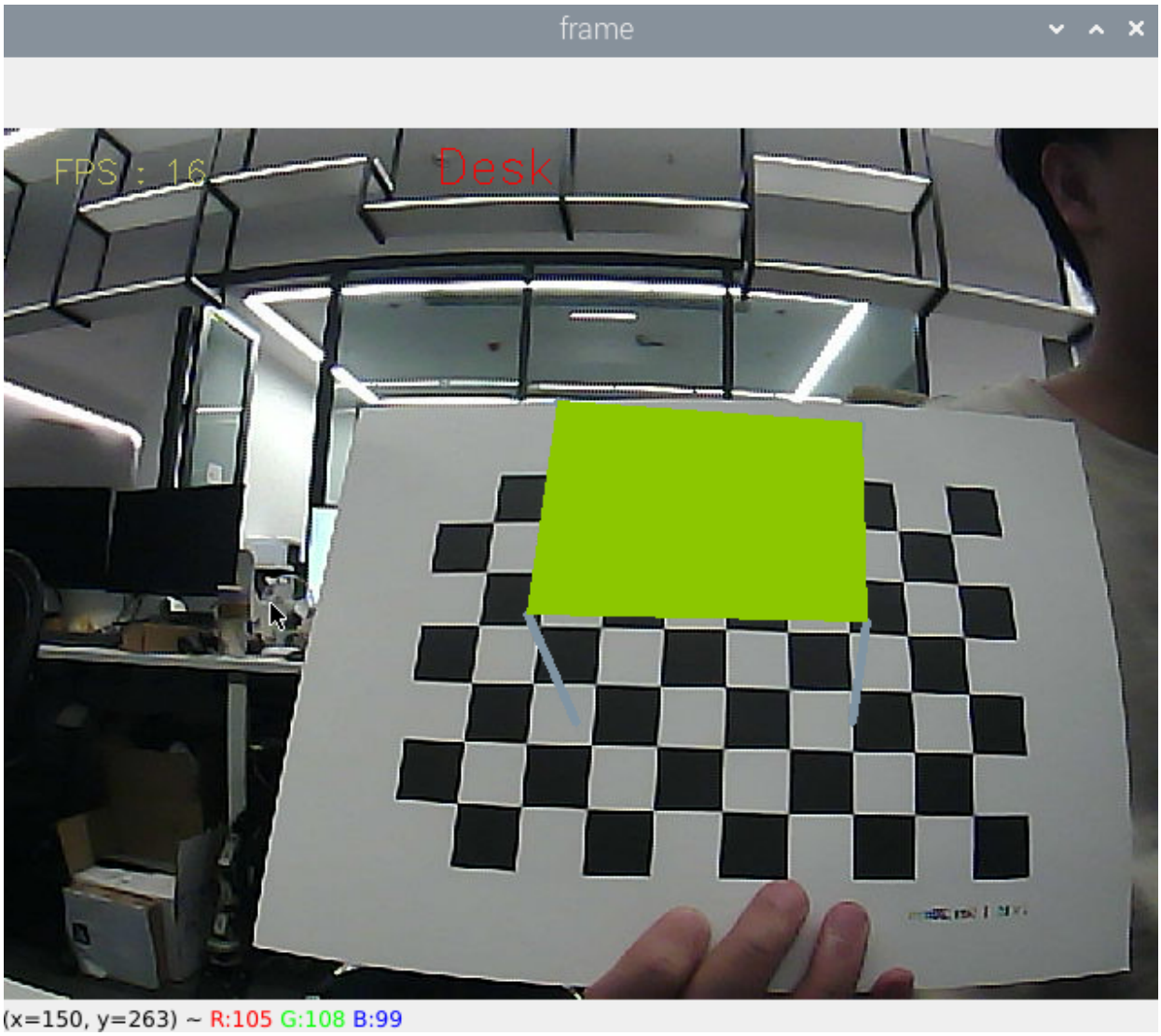
```
ros2 topic list
```




- /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, where the image is posted.

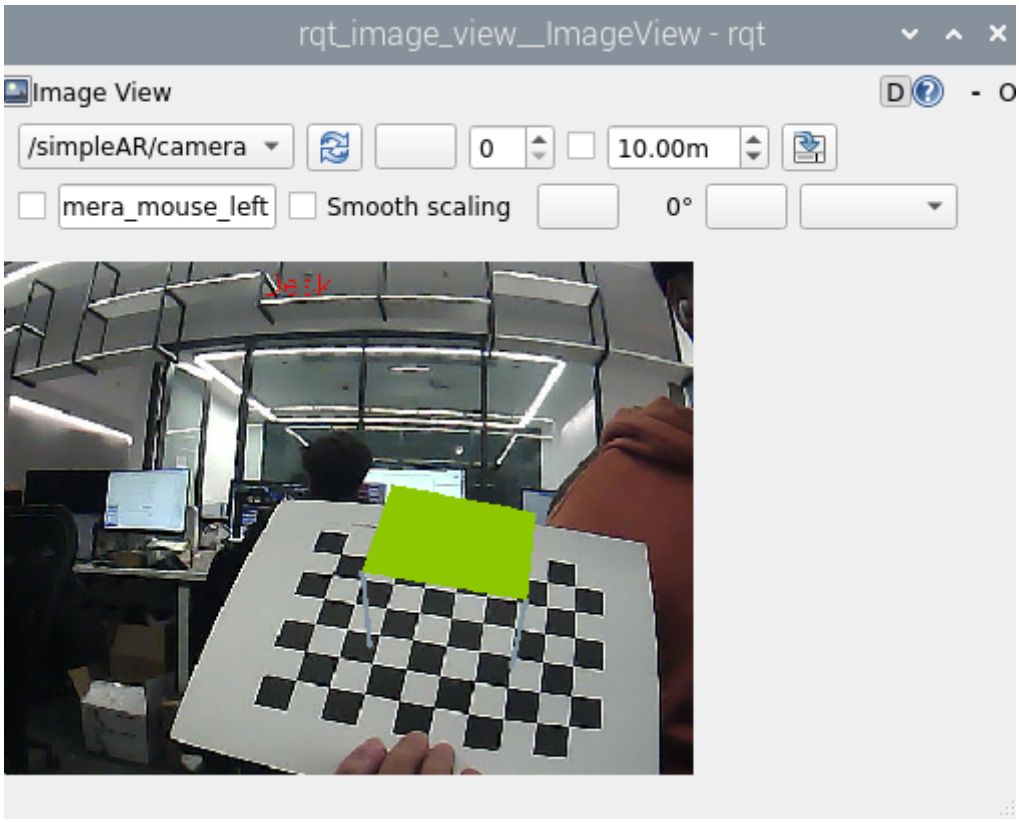
The modification effect can be modified by the following command. For example, I first changed it to Desk and entered the following command in the docker terminal.

```
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. Enter the following command in the docker terminal.

```
ros2 run rqt_image_view rqt_image_view
```




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