

Camera intrinsic calibration

Camera intrinsic calibration

- 2.1、Program Function Description
- 2.2、Preparation before calibration
- 2.3、Program startup
 - 2.3.1, Start the camera
 - 2.3.2, Color Image Calibration
 - 2.3.3, IR infrared calibration

2.1、Program Function Description

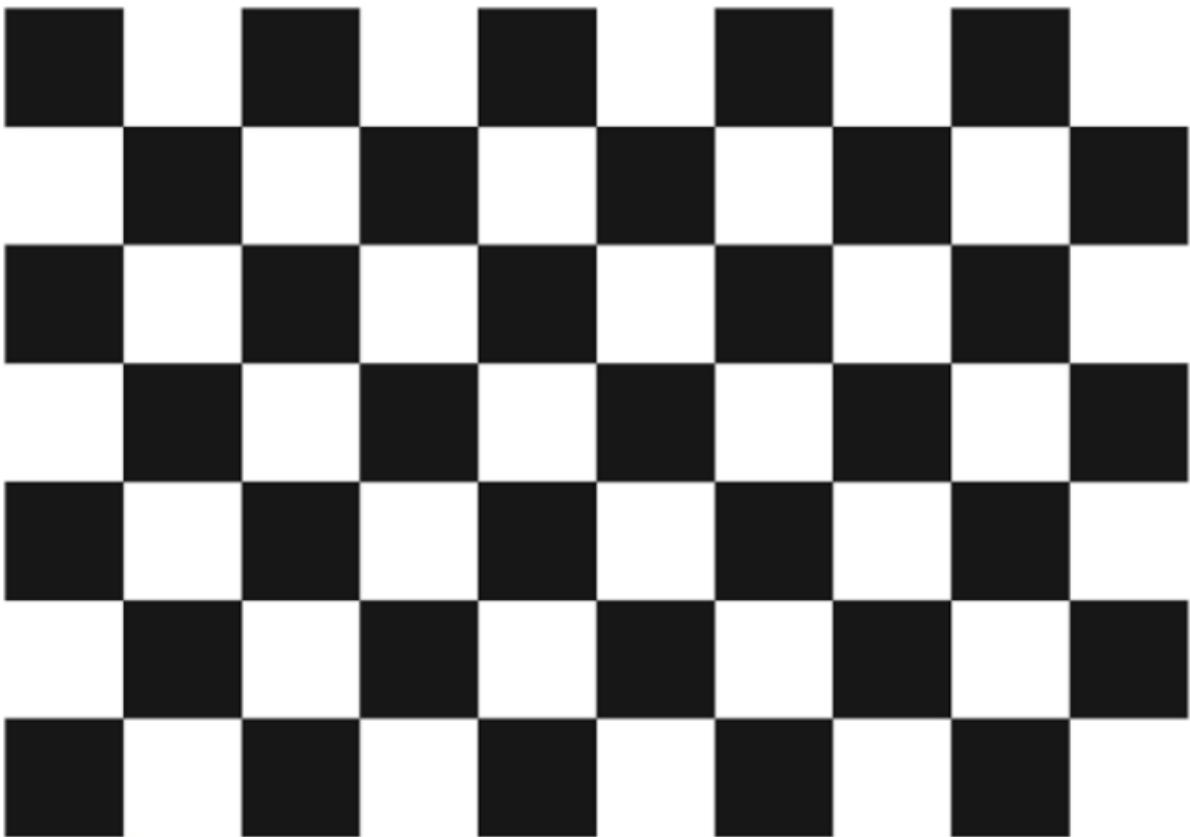
Due to some reasons inside and outside the camera, the image will be greatly distorted, mainly radial deformation and tangential deformation, which will cause the straight line to become curved. The farther the pixel is from the center of the image, the more serious the distortion. In order to avoid the error caused by the data source, the camera parameters need to be calibrated. Calibration is essentially to reversely deduce the inherent and real parameters (internal parameters) of the camera by analyzing the pixels of the taken picture with the help of a known and determined spatial relationship (calibration board).

Wiki: http://wiki.ros.org/camera_calibration

2.2、Preparation before calibration

- A large chessboard of known size. This tutorial uses a 9x6 chessboard and a 20 mm square, which needs to be flattened during calibration. **Calibration uses the internal vertices of the chessboard, so the "10x7" chessboard uses the internal vertex parameter "9x6", as shown in the example below.** Any specification of calibration board can be used, just change the parameters.
- A relatively open area without obstacles and calibration board patterns.
- Monocular camera that publishes images through ROS.

Chessboard (calibration board)



7×10 | Size: 20mm

2.3, Program startup

2.3.1, Start the camera

Take the Astraproplus camera as an example, after SSH connects to the car, enter in the terminal,

```
ros2 launch astra_camera astro_pro_plus.launch.xml
```

Start the camera before calibration, and turn it off after all calibration is completed.

Obi Zhongguang camera models and corresponding launch files:

launch file	camera model
ros2 launch astra_camera astra_pro.launch.xml	Astrapro
ros2 launch astra_camera astro_pro_plus.launch.xml	Astraproplus
ros2 launch astra_camera astra.launch.xml	Astramini

Since the car has no graphical interface, the following calibration program needs to be run in a virtual machine.

Enter in the virtual machine terminal,

```
ros2 topic list
```

```

yahboom@VM: ~
yahboom@VM: ~ 80x24
[ System Information ]
ROS Version: humble
ROS DOMAIN_ID: 99
IP Address: 192.168.2.118
yahboom@VM:~$ ros2 topic list
/camera/color/camera_info
/camera/color/image_raw
/camera/depth/camera_info
/camera/depth/image_raw
/camera/depth/points
/camera/ir/camera_info
/camera/ir/image_raw
/parameter_events
/rosout
/tf
/tf_static
yahboom@VM:~$
```

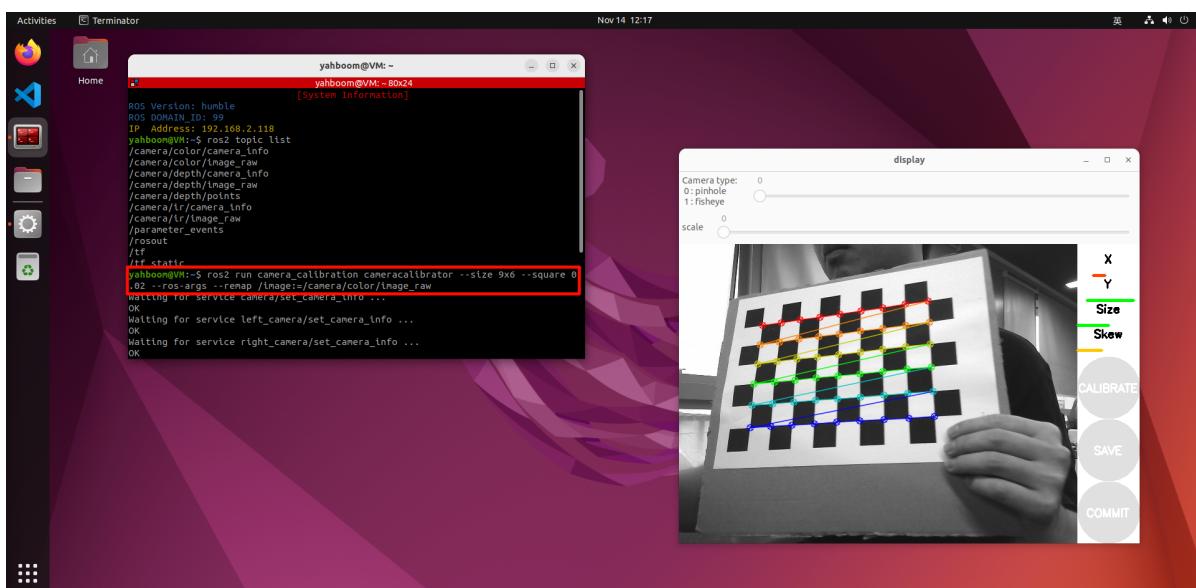
The topic we need to use to calibrate RGB color images is /camera/color/image_raw, and the topic used to calibrate IR infrared images is /camera/ir/image_raw.

2.3.2, Color Image Calibration

Enter in the virtual machine terminal,

```
ros2 run camera_calibration cameracalibrator --size 9x6 --square 0.02 --ros-args
--remap /image:=/camera/color/image_raw
```

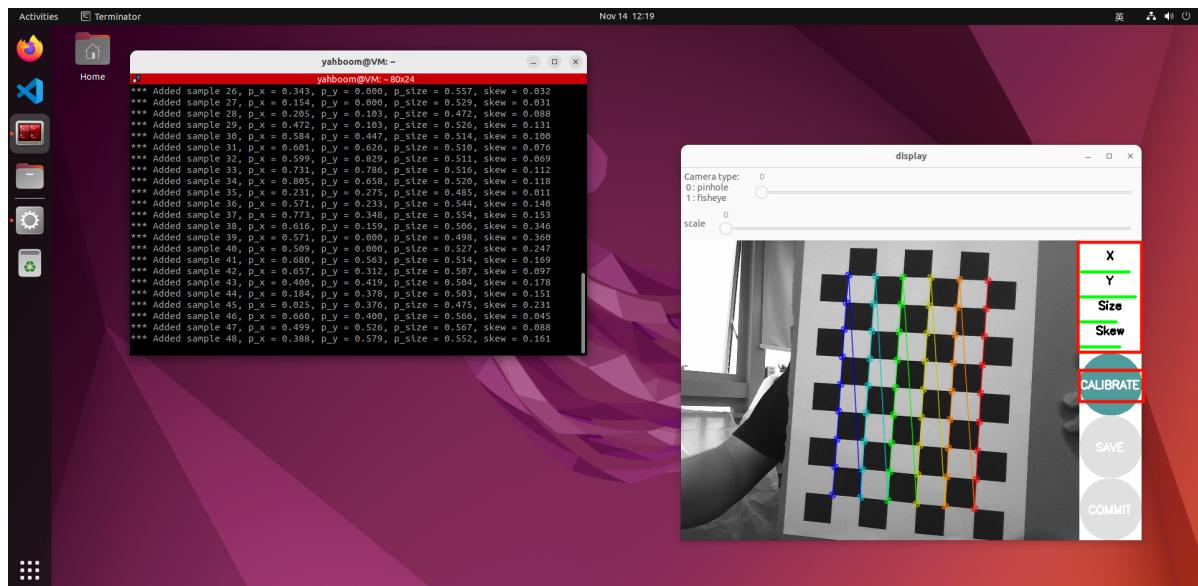
- size: calibrate the number of internal corner points of the chessboard, for example 9X6, the corner points have a total of six rows and nine columns.
- square: the side length of the chessboard, in meters.
- /camera/color/image_raw: RGB image topic name.



- X: Left and right movement of the chessboard in the camera field of view

- Y: Up and down movement of the chessboard in the camera field of view
- Size: Front and back movement of the chessboard in the camera field of view
- Skew: Tilt and rotation of the chessboard in the camera field of view

Collect images by moving up and down, front and back, left and right, and flipping, so that [X], [Y], [Size], and [Skew] on the right turn green, as shown in the figure below, and then click [CALIBRATE] to start calculating the camera internal parameters. The more pictures, the longer it takes, just wait. (Sixty or seventy pictures are enough, too many will easily get stuck).



After calibration, click [SAVE], as shown in the figure below.

```
yahboom@VM: ~
yahboom@VM: ~ 80x24
height
480

[narrow_stereo]

camera matrix
527.255909 0.000000 322.550480
0.000000 528.194975 246.772732
0.000000 0.000000 1.000000

distortion
0.081820 -0.173789 0.005011 0.000092 0.000000

rectification
1.000000 0.000000 0.000000
0.000000 1.000000 0.000000
0.000000 0.000000 1.000000

projection
530.289673 0.000000 322.192138 0.000000
0.000000 532.100647 247.961037 0.000000
0.000000 0.000000 1.000000 0.000000

('Wrote calibration data to', '/tmp/calibrationdata.tar.gz')
```

The calibration results are saved to [/tmp/calibrationdata.tar.gz]. You can move the file out to view the content.

After exiting the program, enter in the virtual machine terminal,

```

#Move the file to the current directory
sudo mv /tmp/calibrationdata.tar.gz .
#Create a folder
mkdir calibrationdata_rgb
#Unzip to the specified folder
tar -zxvf calibrationdata.tar.gz -C ./calibrationdata_rgb

```

```

yahboom@VM:~$ sudo mv /tmp/calibrationdata.tar.gz .
[sudo] password for yahboom:
yahboom@VM:~$ mkdir calibrationdata_rgb
yahboom@VM:~$ tar -zxvf calibrationdata.tar.gz -C ./calibrationdata_rgb
left-0000.png
left-0001.png
left-0002.png
left-0003.png
left-0004.png
left-0005.png
left-0006.png
left-0007.png
left-0008.png
left-0009.png
left-0010.png
left-0011.png
left-0012.png
left-0013.png
left-0014.png
left-0015.png
left-0016.png
left-0017.png
left-0018.png
left-0019.png

```

You will get the calibrated png file, ost.yaml and ost.txt files in the calibrationdata_rgb folder under this directory.

Since the Astra driver loads the built-in parameters calibrated in the code when it is started, there is no need to load this calibrated parameter. The camera internal parameters will be used in the [AR Vision] case in the [OpenCV Series Courses].

2.3.3, IR infrared calibration

Since the RGBD camera uses structured light as the depth imaging principle, the infrared light it projects is a special disordered light spot, which makes the infrared receiving device unable to receive clear and complete picture content.

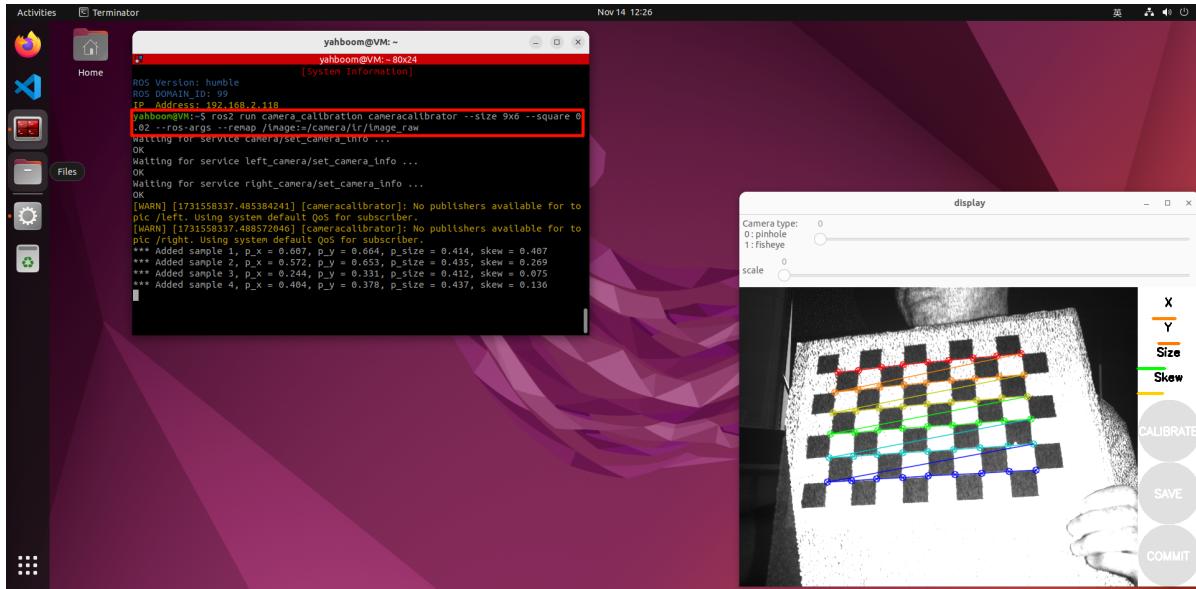
At this time, we can have several special processing methods:

- Force to find various angles and let the camera find the corner point as much as possible (poor accuracy)
- By sticking some frosted translucent paper in front of the red-haired transmitter, the infrared spot is evenly distributed (moderate accuracy, more convenient)
- Block the infrared projection camera and use an external infrared camera for fill light (high accuracy, requires additional equipment)

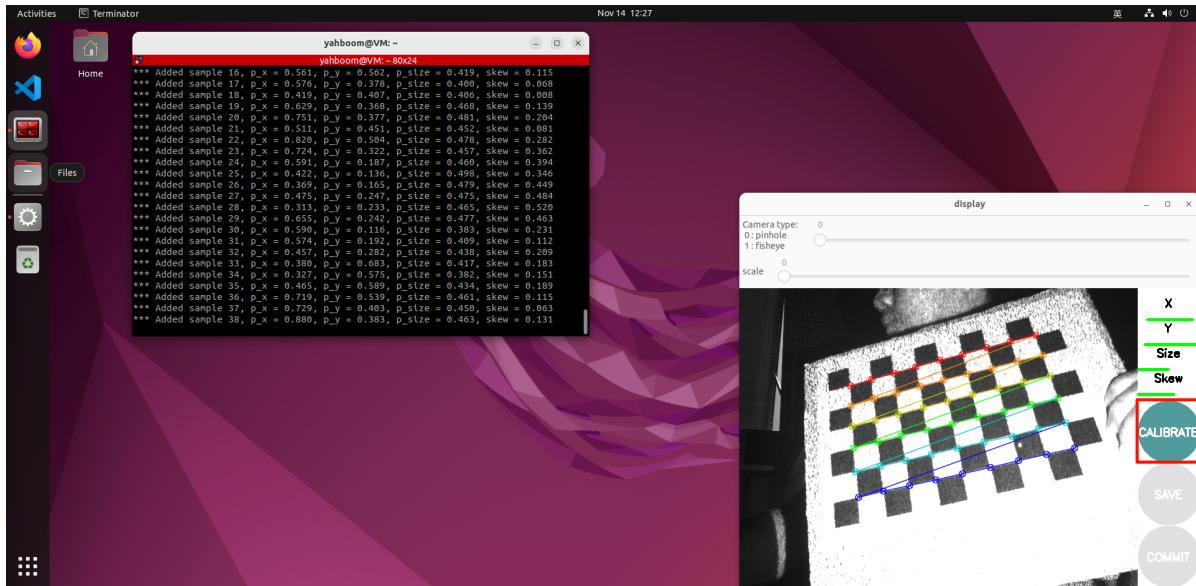
Choose a processing method according to your needs.

Enter in the virtual machine terminal,

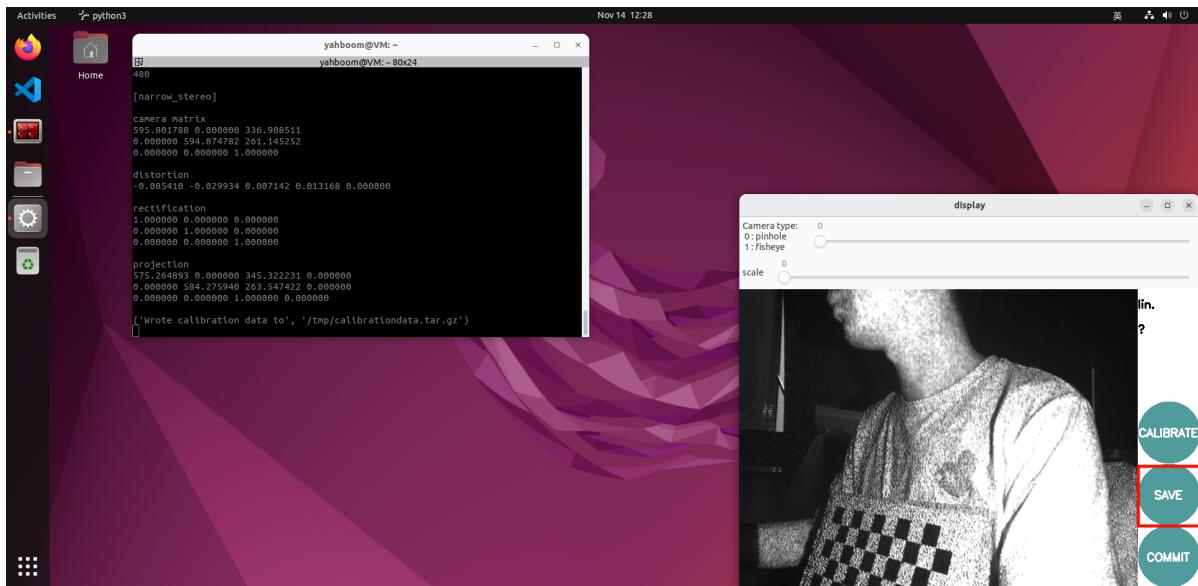
```
ros2 run camera_calibration cameracalibrator --size 9x6 --square 0.02 --ros-args
--remap /image:=/camera/ir/image_raw
```



Similarly, the following operations are similar to the color camera calibration, changing different poses and filling the lines under [X], [Y], [Size], and [Skew] as much as possible.



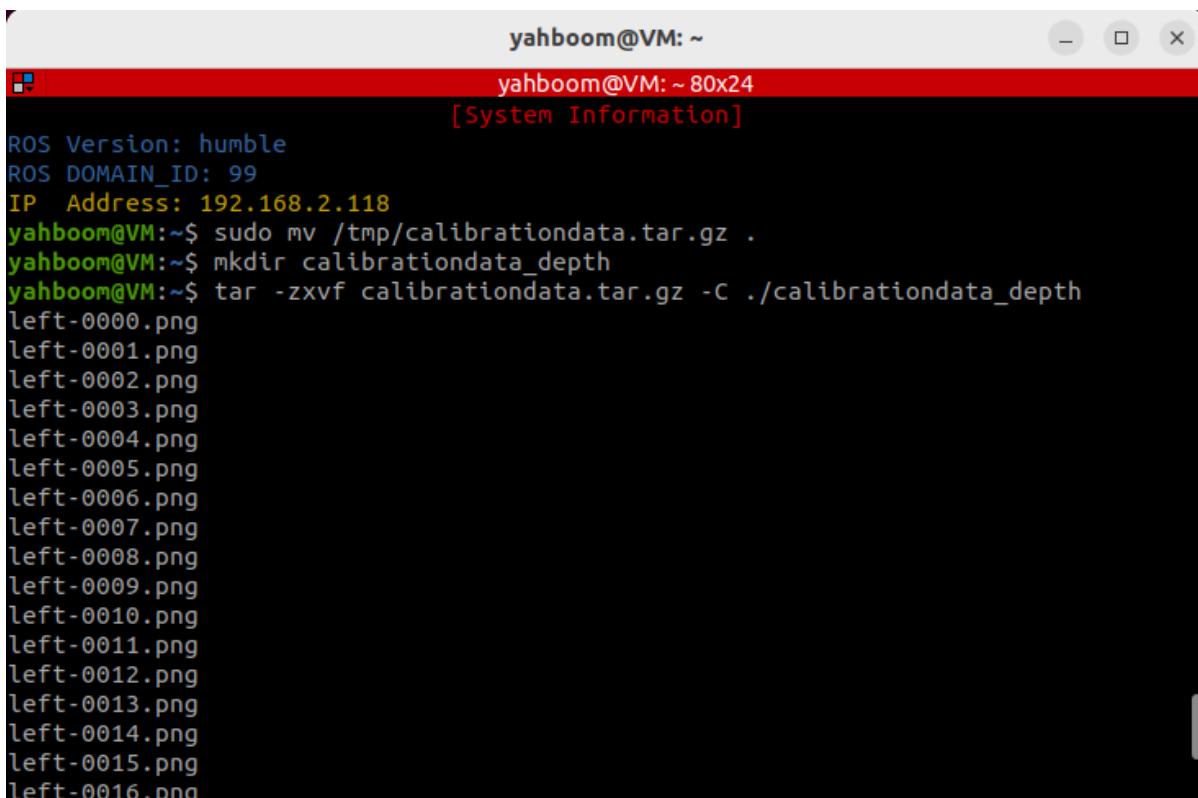
Click [CALIBRATE] to calculate the camera internal parameters, and click [SAVE], as shown in the figure below.



The calibration results are saved to [/tmp/calibrationdata.tar.gz]. You can move the file out to view the content.

After exiting the program, enter in the virtual machine terminal,

```
#Move the file to the current directory
sudo mv /tmp/calibrationdata.tar.gz .
#Create a folder
mkdir calibrationdata_depth
#Unzip to the specified folder
tar -zxvf calibrationdata.tar.gz -C ./calibrationdata_depth
```



You will get the calibrated png file, ost.yaml and ost.txt files in the calibrationdata_depth folder under this directory.