2. Camera internal reference calibration

Before driving the depth camera, you need to be able to recognize the astra camera device in the host; when entering the docker container, you need to mount this astra device in order to be in the docker container and recognize the camera. Matching host has been built environment, do not need additional configuration, if in the new host, you need to add the rules file. The method of adding is very simple, copy the /etc/udev.rules.d/56-orbbecusb.rules file under the host to the /etc/udev.rules.d directory of the new environment, and then reboot once.

1. Pre-calibration preparation

- A large chessboard of known dimensions. This tutorial uses a 9x6 checkerboard grid and 20mm squares, which are spread out for calibration. **Calibration uses the internal vertices of the checkerboard grid, so a "10x7" board uses the internal vertex parameter "9x6", as shown in the example below.** A calibration board of any size is fine, just change the parameters. An empty area with no obstacles or calibration board patterns
- Monocular camera publishing images via ROS

2. Start calibration

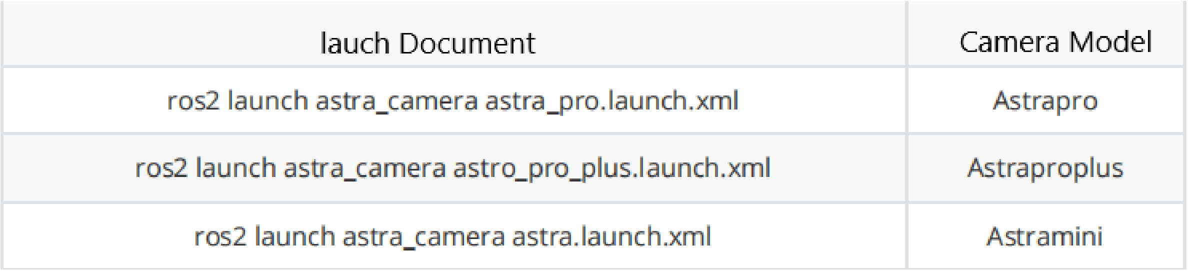
Install the calibration package camera_calibration, docker terminal input.

```
sudo apt install ros-foxy-camera-calibration*
```

Start the camera before calibrating until it is all calibrated and then shut down the camera, start the camera (start the astraproplus camera as an example), and in the Docker terminal enter.

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

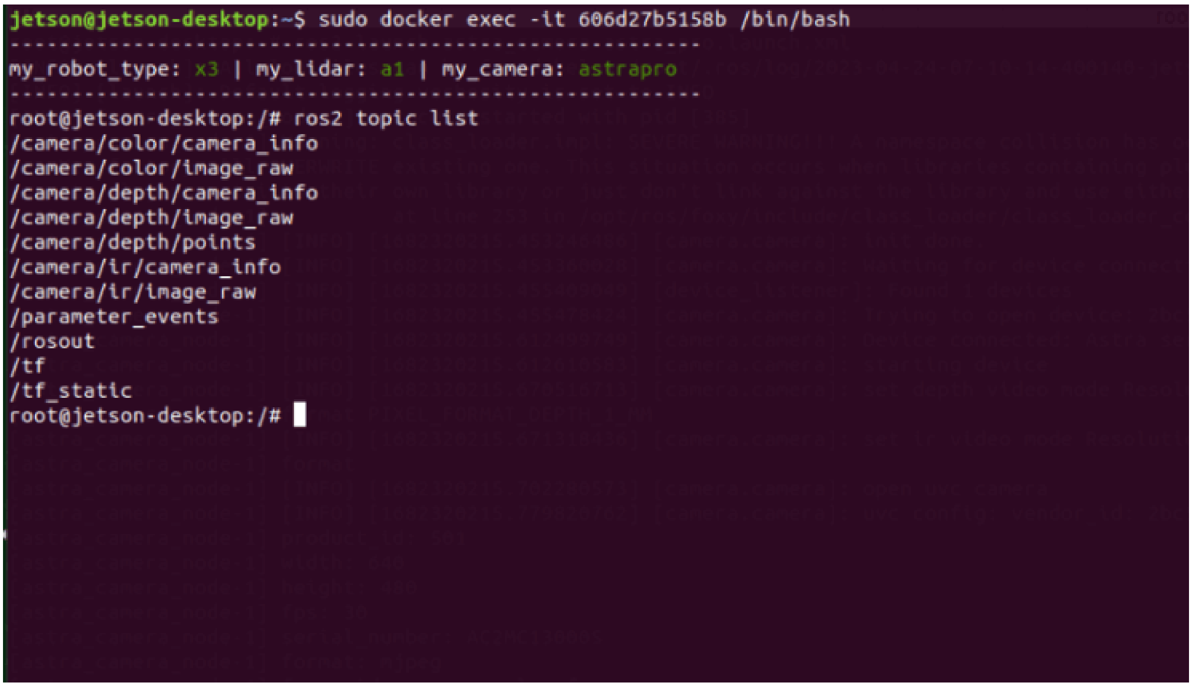
startup reference for other astra camera models.



lauch Document	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

Use the following command to view the topic, entered in the Docker terminal, the

```
ros2 topic list
```



```
jetson@jetson-desktop:~$ sudo docker exec -it 606d27b5158b /bin/bash
---------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
---------------------------------------------------------
root@jetson-desktop:/# 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
root@jetson-desktop:/# 
```

The topic we need for calibrating RGB color images is /camera/color/image_raw.

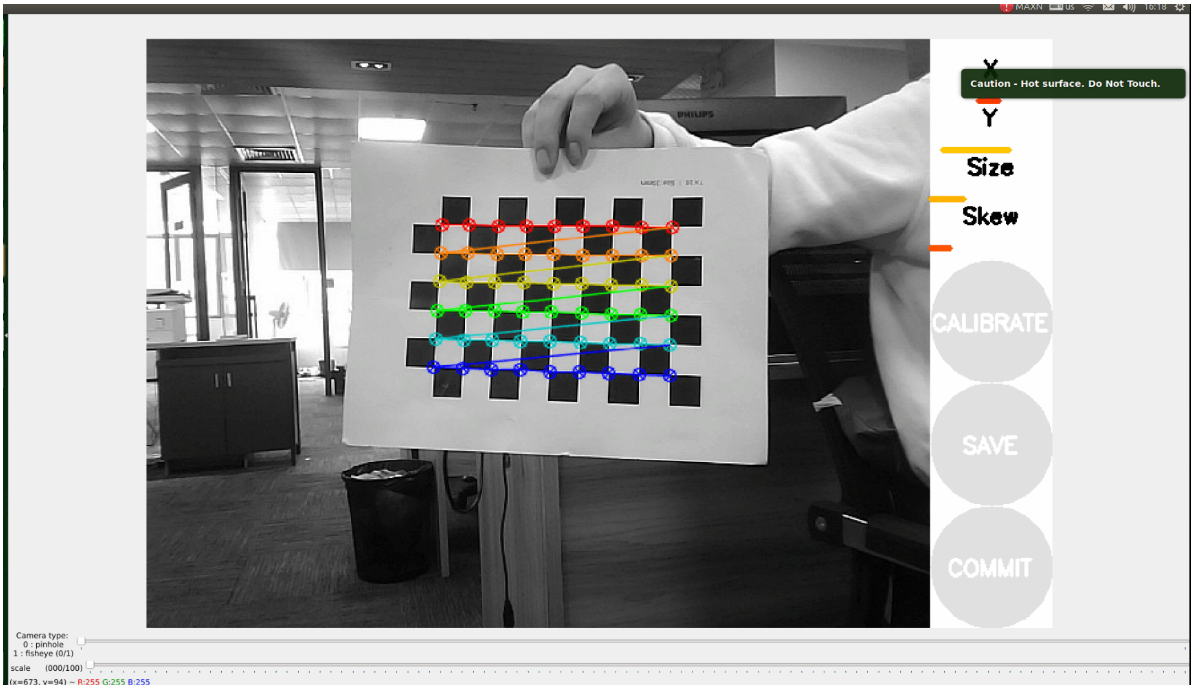
To run the calibrated program, type in the Docker terminal, the

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

size: the number of internal corners of the grid, e.g. 9x6, six rows and nine columns of corners.

square: length of the side of the grid, in meters.

topic name: /camera/color/image_raw, if start **usb_cam**, here change to **/image_raw**.



Caution - Hot surface. Do Not Touch.

Camera type:
0 : pinhole
1 : fisheye (0/1)

scale (000/100)

(x=673, y=94) ~ R:255 G:255 B:255

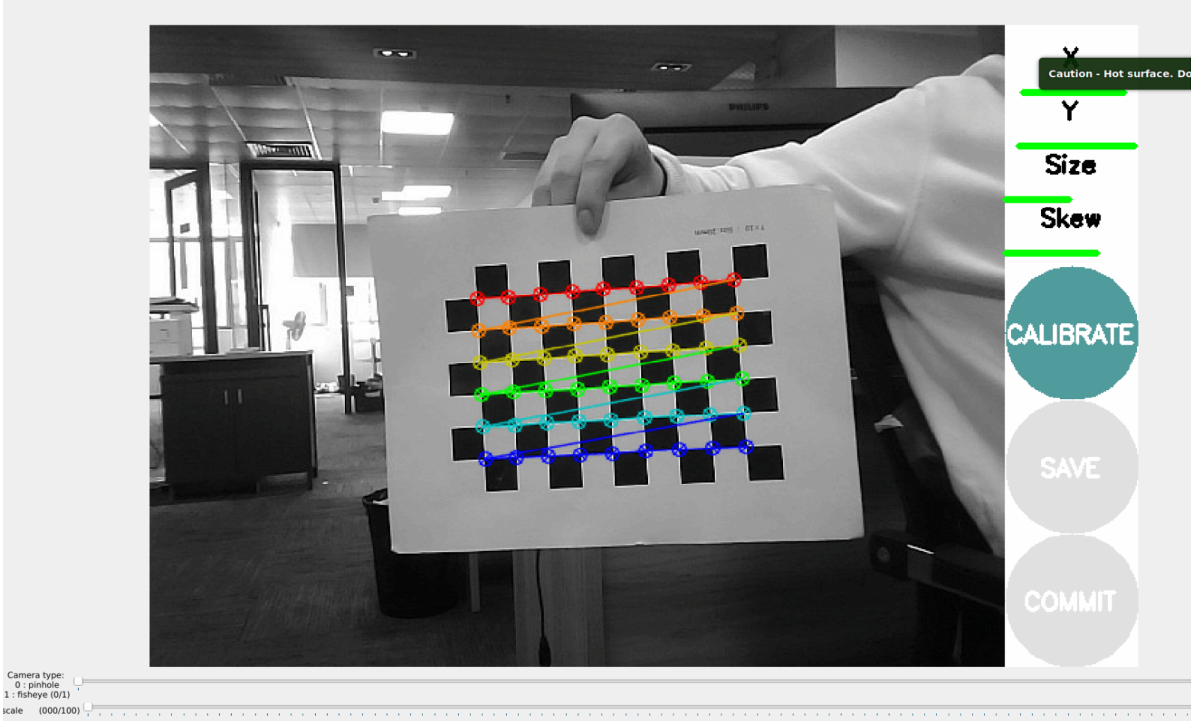
X: left-right movement of the checkerboard in the camera field of view

Y: up and down movement of the checkerboard in the camera's field of view

Size: Back and forth movement of the checkerboard in the camera's field of view.

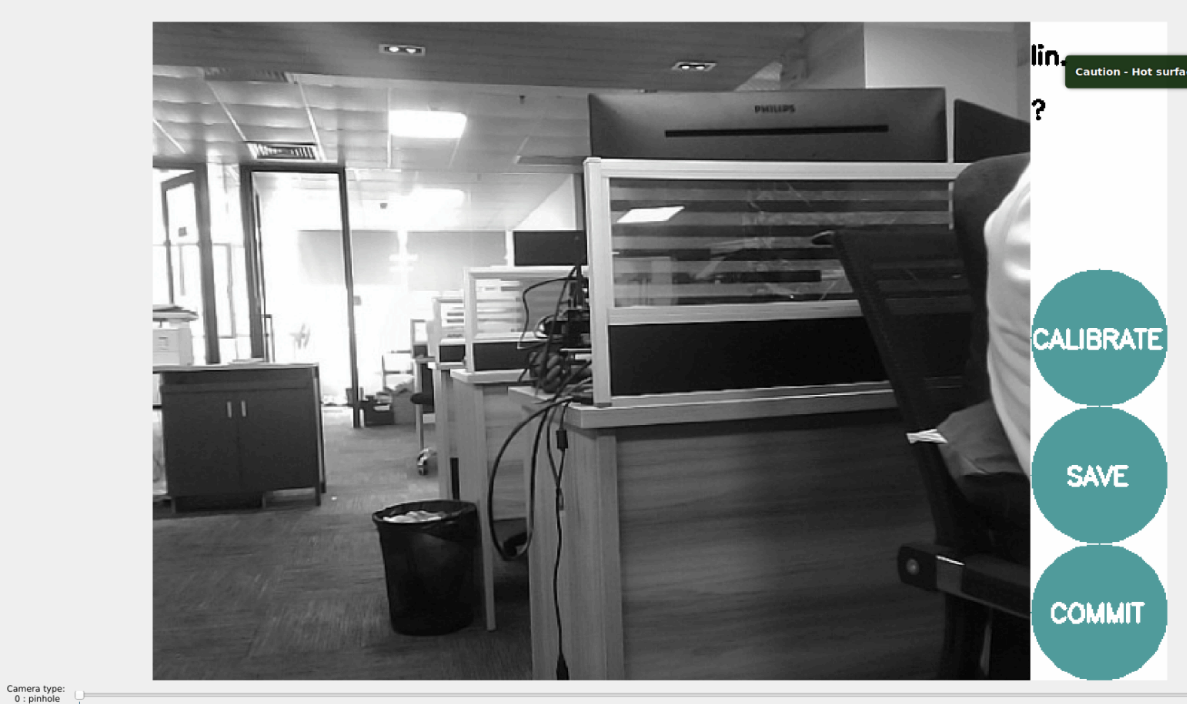
Skew: tilt rotation of the checkerboard grid in the camera view.

As shown in the above figure, you need to capture the image by flipping it up and down, so that the X, Y, Size and Skew on the right side turn green, as shown in the figure below, and then click CALIBRATE to start the calibration.



Camera type:
0 : pinhole
1 : fisheye (0/1)
scale (000/100)

When calibration is complete, click SAVE, as shown below.



Camera type:
0 : pinhole

The calibration results are saved to [/tmp/calibrationdata.tar.gz] in the directory of the terminal where the calibration program is started. After the calibration is finished, you can move out the [/tmp/calibrationdata.tar.gz] file to see the contents.

```
sudo mv /tmp/calibrationdata.tar.gz ~
```

docker terminal input.

```
cd ~
tar -xvf calibrationdata.tar.gz
```

You will get the calibrated png file, ost.yaml and ost.txt file in the terminal directory.

Since the astra driver loads the built-in parameters calibrated in the code when it starts up, it doesn't need to load the calibrated parameters, but it needs to load the parameters when it starts up the usb camera, so you need to replace the parameters with the original built-in parameters after calibration, rename the calibrated ost.yaml to camera_info.yaml, and then replace the original camera_info.yaml, docker terminal input.

```
#先把文件复制到/opt/ros/foxy/share/usb_cam/config
# First copy the file to /opt/ros/foxy/share/usb_cam/config
sudo cp ost.yaml /opt/ros/foxy/share/usb_cam/config
#切换至/opt/ros/foxy/share/usb_cam/config目录下
# Switch to the /opt/ros/foxy/share/usb_cam/config directory
cd /opt/ros/foxy/share/usb_cam/config
#备份原来的camera_info.yaml
# Backup the original camera_info.yaml
sudo mv camera_info.yaml camera_info_BK.yaml
#重命名ost.yaml为camera_info.yaml
# Rename ost.yaml to camera_info.yaml
sudo mv ost.yaml camera_info.yaml
```