Due to some internal and external reasons of the camera, the image will be greatly distorted, mainly radial deformation and tangential deformation, resulting in the bending of the straight line. The farther away from the image center, the more serious the distortion. In order to avoid errors caused by data sources, camera parameters need to be calibrated. In essence, calibration is to use a known and determined spatial relationship (calibration board) to reversely deduce the intrinsic and real parameters (internal parameters) of the camera by analyzing the image pixels taken.

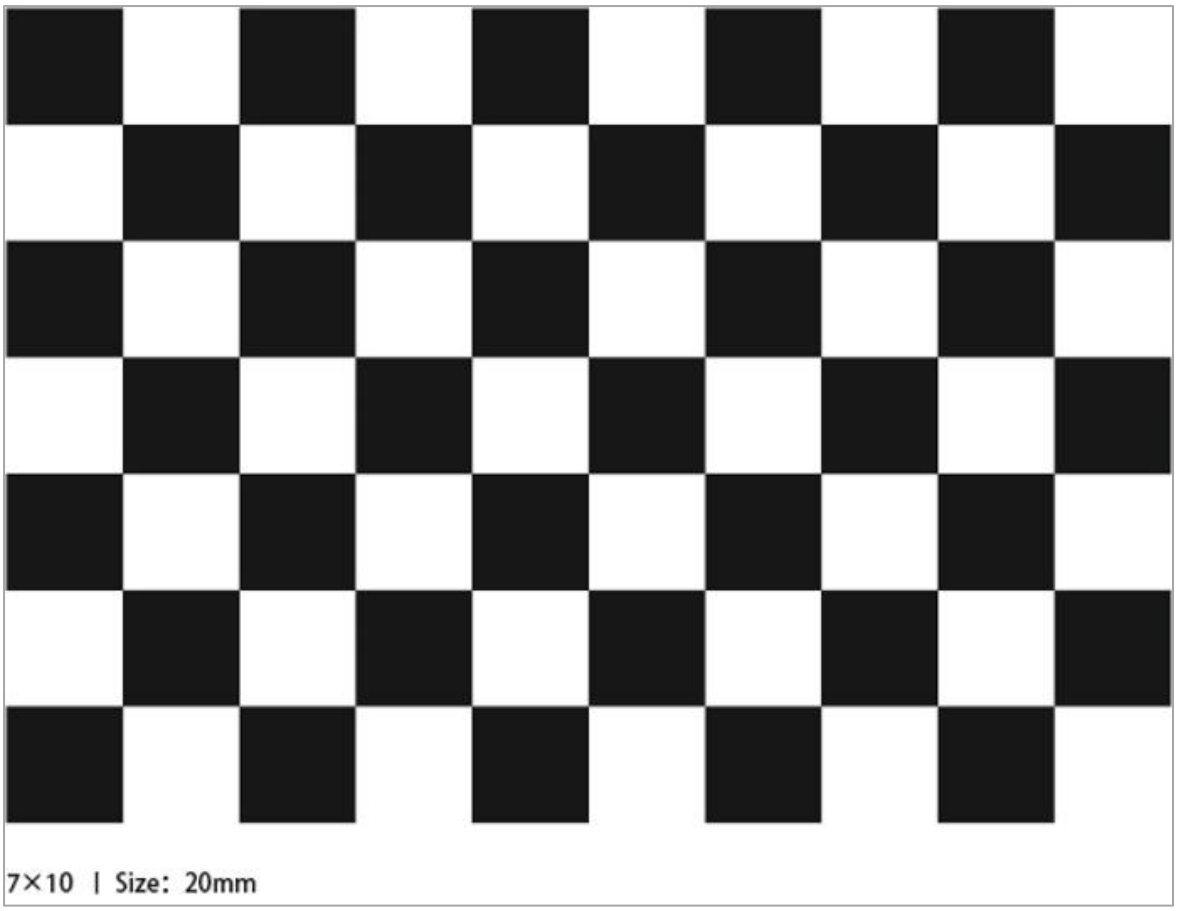
Disadvantages of infrared depth camera ranging:
(1) It is impossible to accurately measure the distance of a black object, because the black object can absorb infrared rays, and the infrared rays cannot return, so it is impossible to measure the distance.
(2) It is impossible to accurately measure the distance of a specular object, because the receiver can only receive the reflected infrared ray when the depth camera is on the mid-vertical line of the specular object, which will lead to overexposure.
(3) It is impossible to accurately measure the distance of transparent objects because infrared rays can pass through transparent objects.
(4) It is impossible to accurately range objects that are too close.

Astra Series

Product Name	ASTRA PRO	ASTRA S	ASTRA
Range	0.6m – 8m	0.4m – 2m	0.6m – 8m
FOV	60°H x 49.5°V x 73°D		
RGB Image Res.	1280 x 720 @30fps	640 x 480 @30fps	
Depth Image Res.	640 x 480 @30fps		
Size	165mm x 30mm x 40mm		
Temperature	0 – 40°C		
Power Supply	USB 2.0		
Power Consumption	< 2.4 W		
Operating Systems	Android/Linux/Windows 7/8/10		
SDK	Astra SDK or OpenNI		
Microphones	2 (Built – in)		

1.1 Preparation before calibration

A large chessboard of known size. This tutorial uses a 9x6 checkerboard and a 20mm square, which should be flattened during calibration.



Astra camera model and corresponding launch file.

Launch File	Camera model
astra.launch	Astra，Astra S，Astra mini，Astra mini S
astraplus.launch	Astra plus
astrapro.launch	Astra pro
embedded_s.launch	Deeyea
dabai_u3.launch	Dabai
gemini.launch	Gemini

Input following command to view device.

```
lsusb
```

```
jetson@jetson-yahboom:~$ lsusb
Bus 002 Device 002: ID 0bda:0411 Realtek Semiconductor Corp.
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 8087:0a2b Intel Corp.
Bus 001 Device 014: ID 2bc5:0403
Bus 001 Device 013: ID 2bc5:0501
Bus 001 Device 012: ID 05e3:0608 Genesys Logic, Inc. Hub
Bus 001 Device 002: ID 0bda:5411 Realtek Semiconductor Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Depth camera ID: 2bc5:0403 Color camera ID: 2bc5:0501
These two IDs indicate that the device is connected.

2.Start Astra camera
Input following command to start Astra camera.

```
roslaunch astra_camera astrapro.launch
```

Input following command to view image topic.

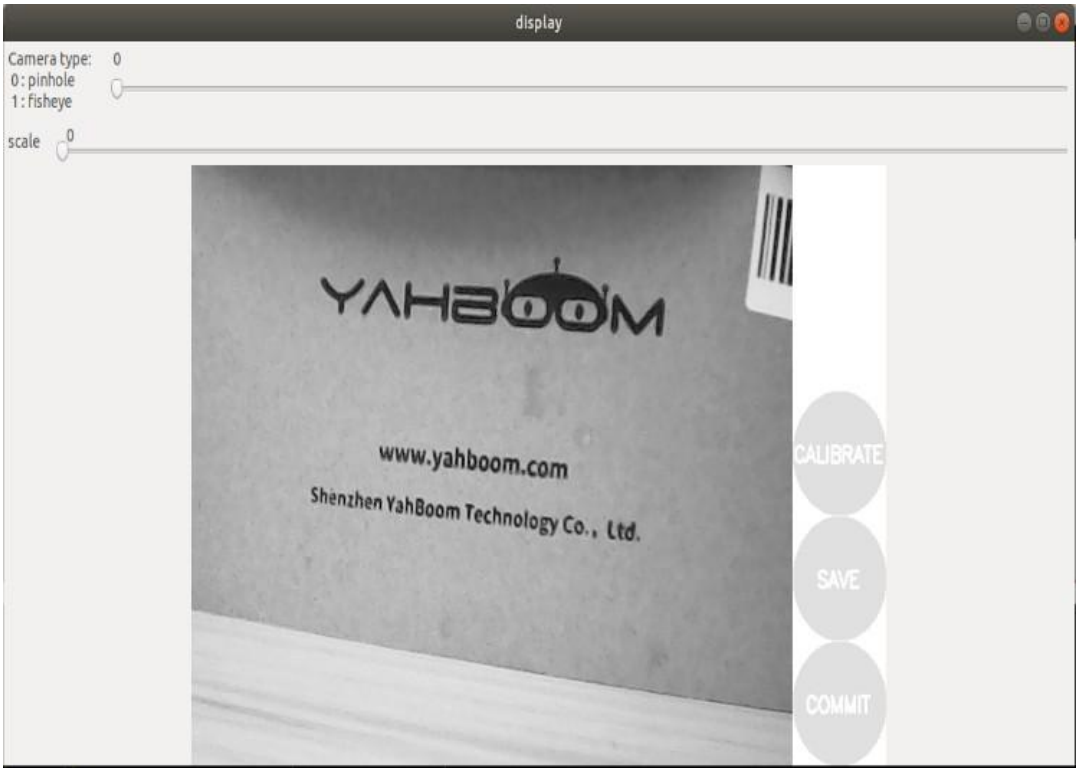
```
rostopic list
```

```
/camera/rgb/camera_info
/camera/rgb/image_raw
/camera/rgb/image_raw/compressed
/camera/rgb/image_raw/compressed/parameter_descriptions
/camera/rgb/image_raw/compressed/parameter_updates
/camera/rgb/image_raw/compressedDepth
/camera/rgb/image_raw/compressedDepth/parameter_descriptions
/camera/rgb/image_raw/compressedDepth/parameter_updates
/camera/rgb/image_raw/theora
/camera/rgb/image_raw/theora/parameter_descriptions
/camera/rgb/image_raw/theora/parameter_updates
/camera/rgb/image_rect_color
/camera/rgb/image_rect_color/compressed
/camera/rgb/image_rect_color/compressed/parameter_descriptions
/camera/rgb/image_rect_color/compressed/parameter_updates
/camera/rgb/image_rect_color/compressedDepth
/camera/rgb/image_rect_color/compressedDepth/parameter_descriptions
/camera/rgb/image_rect_color/compressedDepth/parameter_updates
/camera/rgb/image_rect_color/theora
/camera/rgb/image_rect_color/theora/parameter_descriptions
/camera/rgb/image_rect_color/theora/parameter_updates
/camera/rgb_rectify_color/parameter_descriptions
/camera/rgb_rectify_color/parameter_updates
/rosout
/rosout_agg
/tf_static
```

Input following command to start calibration node

```
# Color chart
rosrun camera_calibration cameracalibrator.py image:=/camera/rgb/image_raw
camera:=/camera/rgb --size 9x6 --square 0.02
```

Size: the number of internal corners of the checkerboard, such as 9X6, with six rows and nine columns of corners.
Square: The side length of the checkerboard, in meters.
Image and camera: set the image topic published by the camera.



3. Color map calibration

Calibration interface

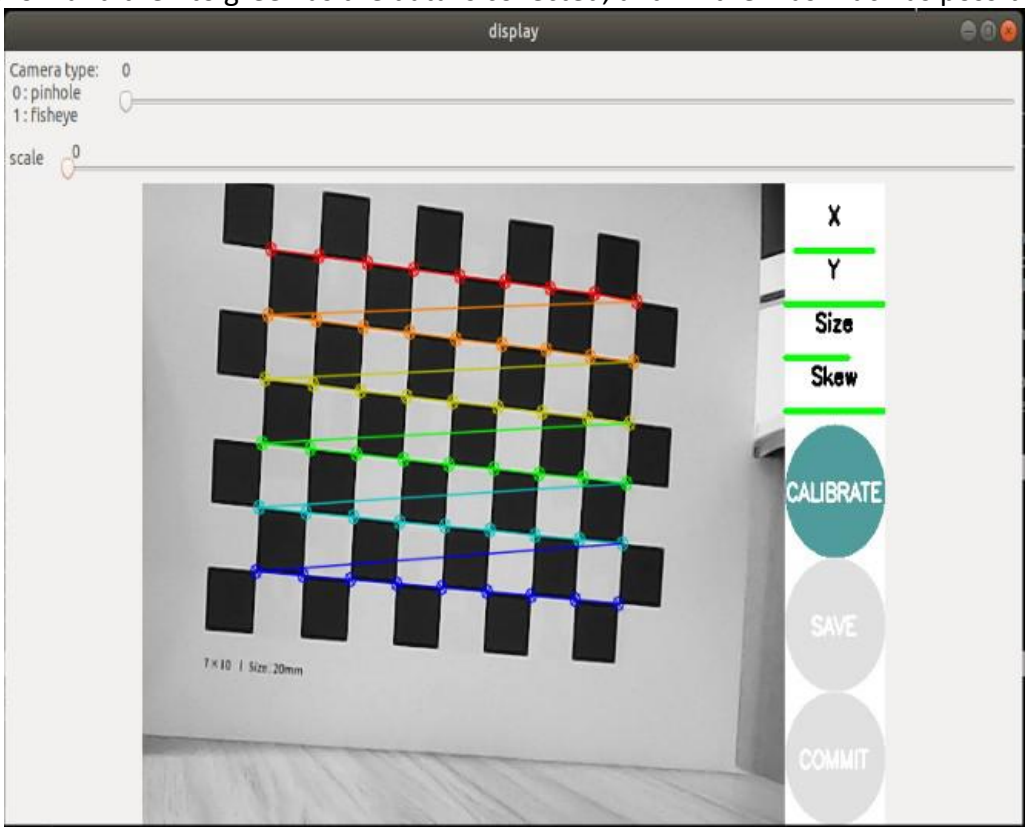
X: Left and right movement of checkerboard in camera view.

Y: Move the checkerboard up and down in the view of the camera.

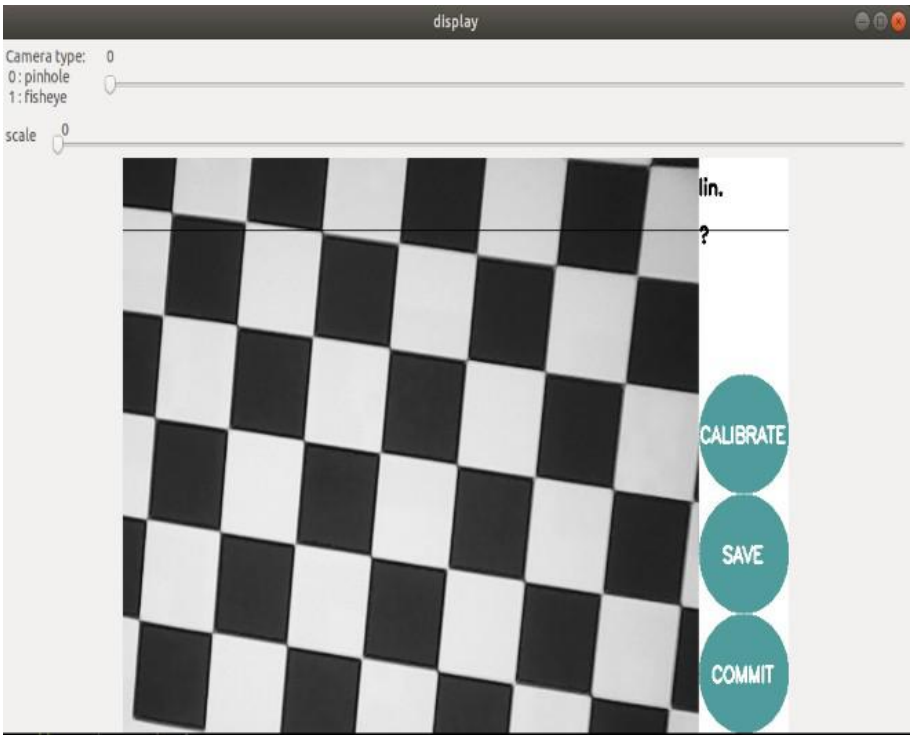
Size: The forward and backward movement of the checkerboard in the camera field of view.

Skew: The tilt and rotation of the checkerboard in the camera field of view.

After successful start up, put the checkerboard in the center of the screen and change different positions. The system will identify itself. The best case is that the lines under [X], [Y], [Size] and [Skew] will change from red to yellow and then to green as the data is collected, and fill them as much as possible.



Click [CALIBRATE] to calculate the camera internal parameters. The more pictures, the longer the time, and the better to wait.
(60 or 70PCS is OK, too many are easy to get stuck)



Click [SAVE] to save the un-calibrated result. The bottom line appears. Click [COMMIT] to exit.

```
**** Calibrating ****
D = [-0.07028213194362816, 0.00043818252903837866, -0.012450845172241 07, 0.000404835
0406427093, 0.0]
K = [543.8333273852593, 0.0, 344.1989291964055, 0.0, 544.5128476949725, 219.77155460
528877, 0.0, 0.0, 1.0]
R = [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
P = [530.847900390625, 0.0, 345.7782327897439, 0.0, 0.0, 534.5553588867188, 214.9890
440488889, 0.0, 0.0, 0.0, 1.0, 0.0]
None
# oST version 5.0 parameters


[image]

width
640

height
480

[narrow_stereo]

camera matrix
543.833327 0.000000 344.198929
0.000000 544.512848 219.771555
0.000000 0.000000 1.000000

distortion
-0.070282 0.000438 -0.012451 0.000405 0.000000

rectification
1.000000 0.000000 0.000000
0.000000 1.000000 0.000000
0.000000 0.000000 1.000000

projection
530.847900 0.000000 345.778233 0.000000
0.000000 534.555359 214.989044 0.000000
0.000000 0.000000 1.000000 0.000000

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

After calibration, the calibration result is in [/tmp/calibrationdata. tar. gz]. We can enter the following command to move the file to view its contents.

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

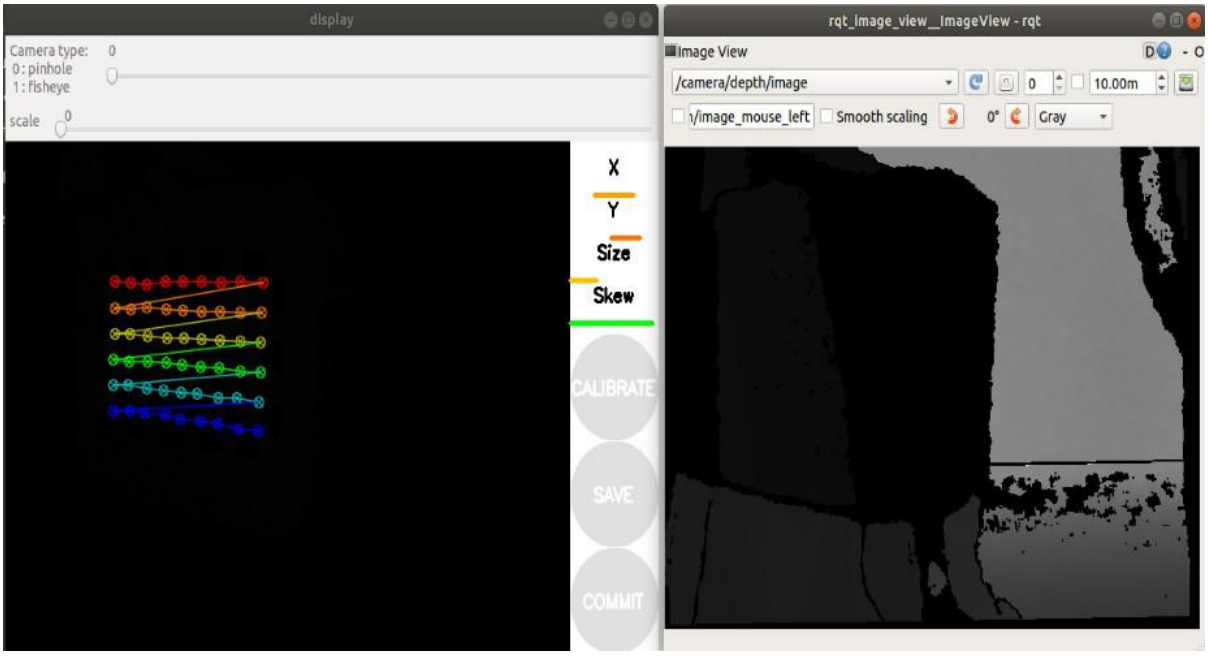
After extrat, there are the image just calibrated, an ost.txt file and an ost.yaml file. The yaml file is what we need, but it needs to be modified before it can be used.
Change ost.yaml to head_ camera.yaml
Add camera_ Name: Change the following name to head_ Camera moves files to~/. ros/camera_ Under info folder.

4. Depth Chart Setting
Start calibration node

```
rosrun camera_calibration cameracalibrator.py image:=/camera/ir/image
camera:=/camera/ir --size 9x6 --square 0.02
```

Size: Calibrate the number of internal corner points of the chessboard, such as 9X6, with a total of six rows and nine columns of corner points.
Square: The side length of a checkerboard, in meters.
Image and camera: Set the image topic posted by the camera.
The calibration process is similar to color camera calibration, but the visualization of depth camera calibration is not so intuitive; "The calibration results are only displayed during recognition, and when not recognized, it becomes pitch black.". It is difficult to calibrate. "We can start the depth image again to view the content of the screen, and if we can't see it clearly, we can open a color image again.".

```
rqt_image_view
```



The following operations are similar to color camera calibration, changing different poses. The system will recognize it independently. The best situation is that the lines under [X], [Y], [Size], and [Skew] will first change from red to yellow and then green as the data is collected, and fill them up as much as possible. click

【CALIBRATE】Calculate the internal parameters of the camera, the more pictures, the longer the time, just wait. (Sixty or seventy sheets are enough, too many are easy to get stuck) Click [SAVE] to save the uncalibrated results, and the bottom line appears, click [COMMIT] to exit.

After the calibration is over, the calibration result exists in [/tmp/calibrationdata.tar.gz], you can move it out to see the content

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

After decompression, there are the pictures just calibrated, an ost.txt file and an ost.yaml file. The yaml file is what we need, but it needs to be modified before it can be used.
Change ost.yaml to depth_Astra_Orbbec.yaml
Change the name behind camera_name: to depth_Astra_Orbbec and move the file to the ~/.ros/camera_info folder