2 Astra camera calibration

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

Official website link: https://orbbec3d.com/develop/

Astra Camera: https://github.com/orbbec/ros_astra_camera

Developer Community: https://developer.orbbec.com.cn/download.html?id=53

Due to some reasons inside and outside the camera, the image will be greatly distorted, mainly radial deformation and tangential deformation, resulting in the bending of the straight line. The farther the pixel is from the center of the image, the more serious the distortion. In order to avoid errors caused by the data source, it is necessary to calibrate the parameters of the camera. The calibration essentially uses a known and determined spatial relationship(calibration board) to reversely deduce the inherent and real parameters of the camera(internal parameters) by analyzing the pixels of the photographed pictures.

Disadvantages of IR depth camera ranging:

(1) It is impossible to accurately measure the distance of black objects, because black materials can absorb infrared rays, and the infrared rays cannot return, so the distance measurement cannot be performed.

(2) It is impossible to accurately measure the distance of the mirror reflection object, because only when the depth camera is on the mid-perpendicular line of the mirror object, the receiver can receive the reflected infrared rays, 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 measure the distance of objects that are too close. The principle is omitted

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)		

2.1 Preparation before calibration

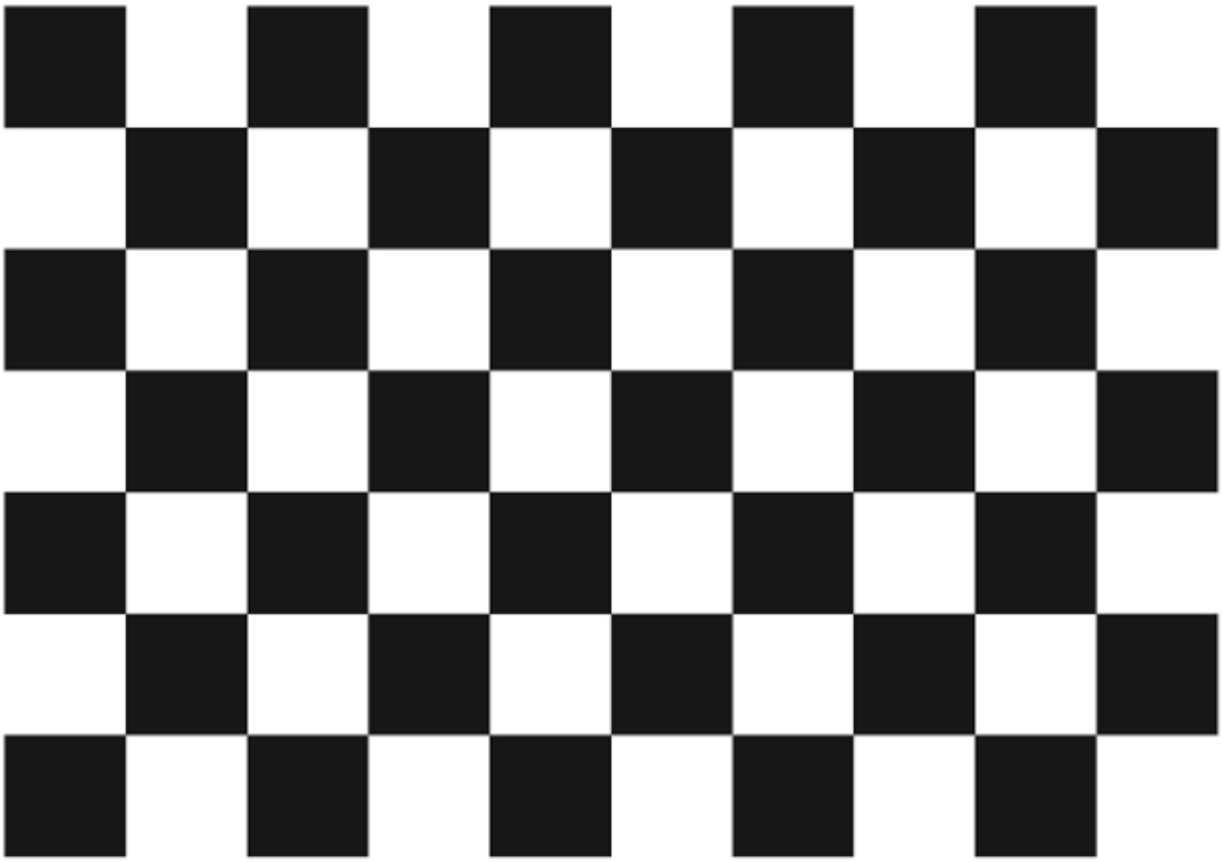
- A large chessboard . This tutorial uses a 9x6 checkerboard and 20mm squares, which should be flattened for calibration.

  The calibration uses the interior vertices of the checkerboard, so a "10x7" checkerboard uses the interior vertex parameter "9x6" as shown in the example below.

  Any specification of the calibration board can be used, just change the parameters.

- A little more open area, no obstacles and no calibration plate pattern
- Monocular camera that publishes images via ROS

Checkerboard(calibration board)



7×10 | Size: 20mm

Aobi Zhongguang camera model and the corresponding launch file

Launch file	Start the camera model
astra.launch	Astra, Astra S, Astra mini, Astra mini S
astraplus.launch	more
astrapro.launch	Astra pro
embedded_s.launch	Deeyea
dabai_u3.launch	Nature
gemini.launch	Gemini

device view

```
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 appear, indicating that the device is connected.

2.2 Astra calibration

Turn on the camera before calibration and turn off the camera until all calibrations are complete.

Start Astra Camera

```
roslaunch yahboomcar_visual astra_calibration.launch
```

This startup command includes an IR image conversion node. The conversion is because the IR infrared camera is viewing a 16-bit image when it is calibrated, and the image cannot be clearly seen. It is necessary to normalize the 16-bit into a value range of 0-255. 8-bit picture, so you can see it clearly.

View Image Topics

```
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
```

2.2.1 Color icon setting

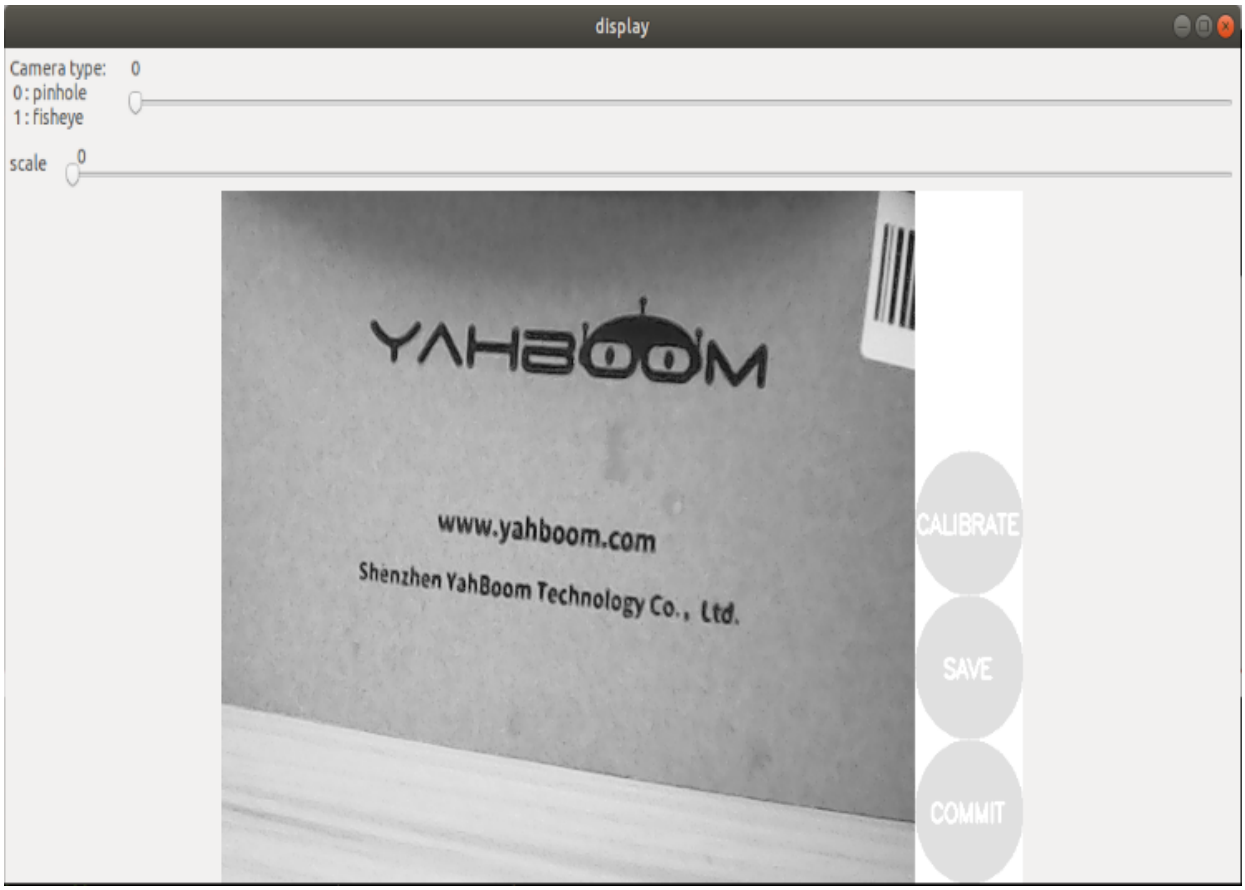
Start the calibration node

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

size: The number of internal corner points of the calibration checkerboard, such as 9X6, the corner points have a total of six rows and nine columns.

square: The side length of the checkerboard, in meters.

image and camera: Set the topic of images posted by the camera.



Calibration interface

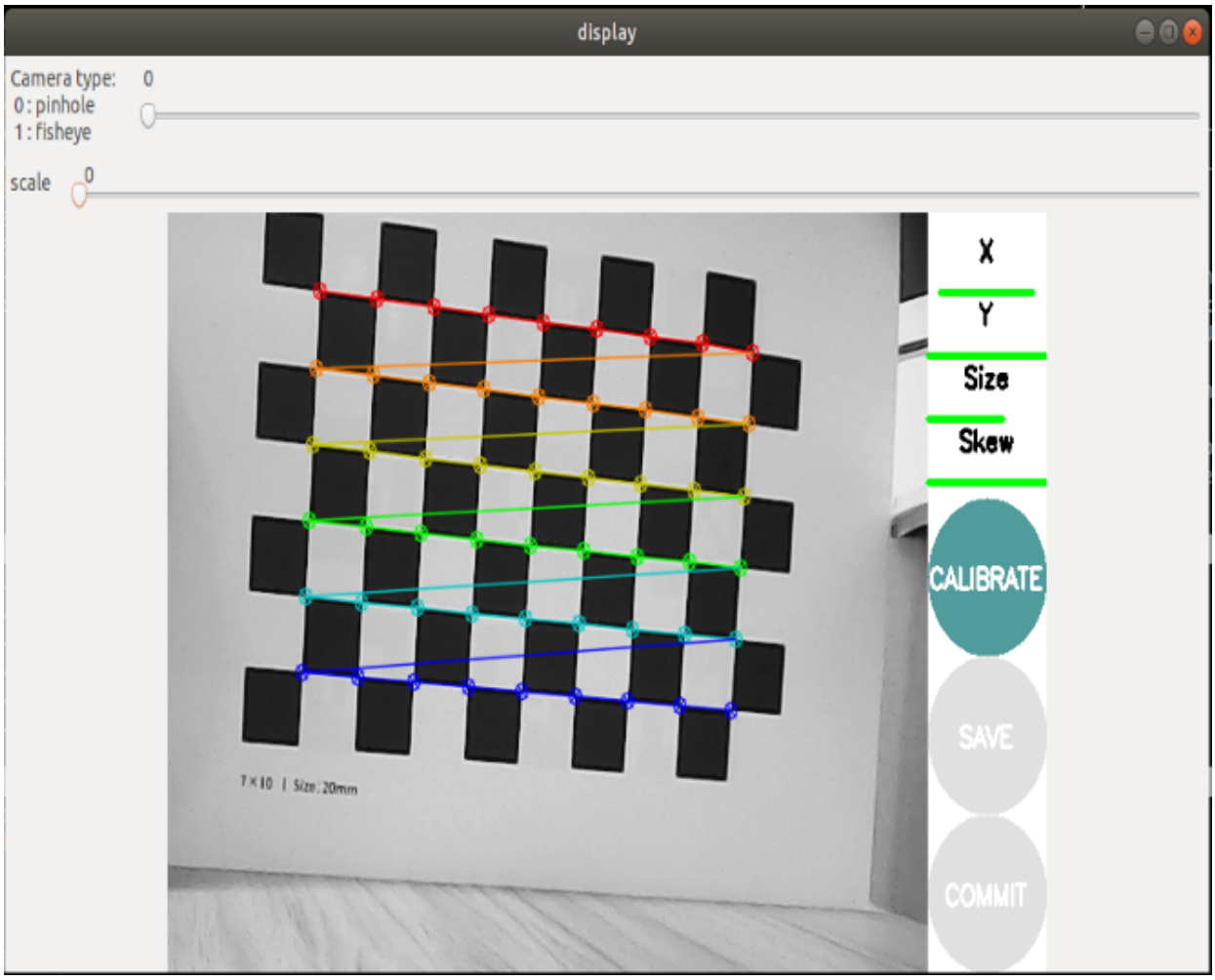
X: left and right movement of the checkerboard in the camera's field of view

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

Size : The back and forth movement of the checkerboard in the camera's field of view

Skew: Tilt rotation of the checkerboard in the camera's field of view

After the startup is successful, put the checkerboard in the center of the screen and change different poses. The system will recognize it autonomously. In the best case, the lines under [X], [Y], [Size], and [Skew] first turn from red to yellow and then green as the data is collected, filling as much as possible.



- Click [CALIBRATE] to calculate the internal parameters of the camera. The more pictures, the longer the time, just wait. (Sixty or seventy sheets are almost the same, too many are easy to get stuck).
- Click [SAVE] to save the calibration results to [/tmp/calibrationdata.tar.gz].
- Click [COMMIT] to write the calibration file to the [.ros/camera_info/rgb_Astra_Orbbec.yaml] file. The calibration result is automatically read the next time the camera is turned on.

```
**** Calibrating ****
D = [-0.07028213194362816, 0.0004381825290383 7866, -0.01245084517224107, 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 the calibration, you can move out the [/tmp/calibrationdata.tar.gz] file to see the content

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

After decompression, it contains the image just calibrated, an ost.txt file and an ost.yaml file.

2.2.2 ir infrared calibration

After dealing with the problem of data normalization, there will still be problems. Because RGBD cameras, cameras with structured light as the principle of depth imaging, project infrared light as a special disordered spot, so that the infrared receiving device cannot receive clear and complete signals. screen content.

At this point we can have several special processing methods:

- Forcibly find various angles, and let the camera find the corners as much as possible(poor accuracy)
- By sticking some frosted translucent paper in front of the red hair emitter, spread the infrared spot evenly(moderate precision, more convenient)
- Block the infrared projection camera and use an external infrared camera to fill in the light(high precision, additional equipment is required)

Choose the treatment method according to your needs.

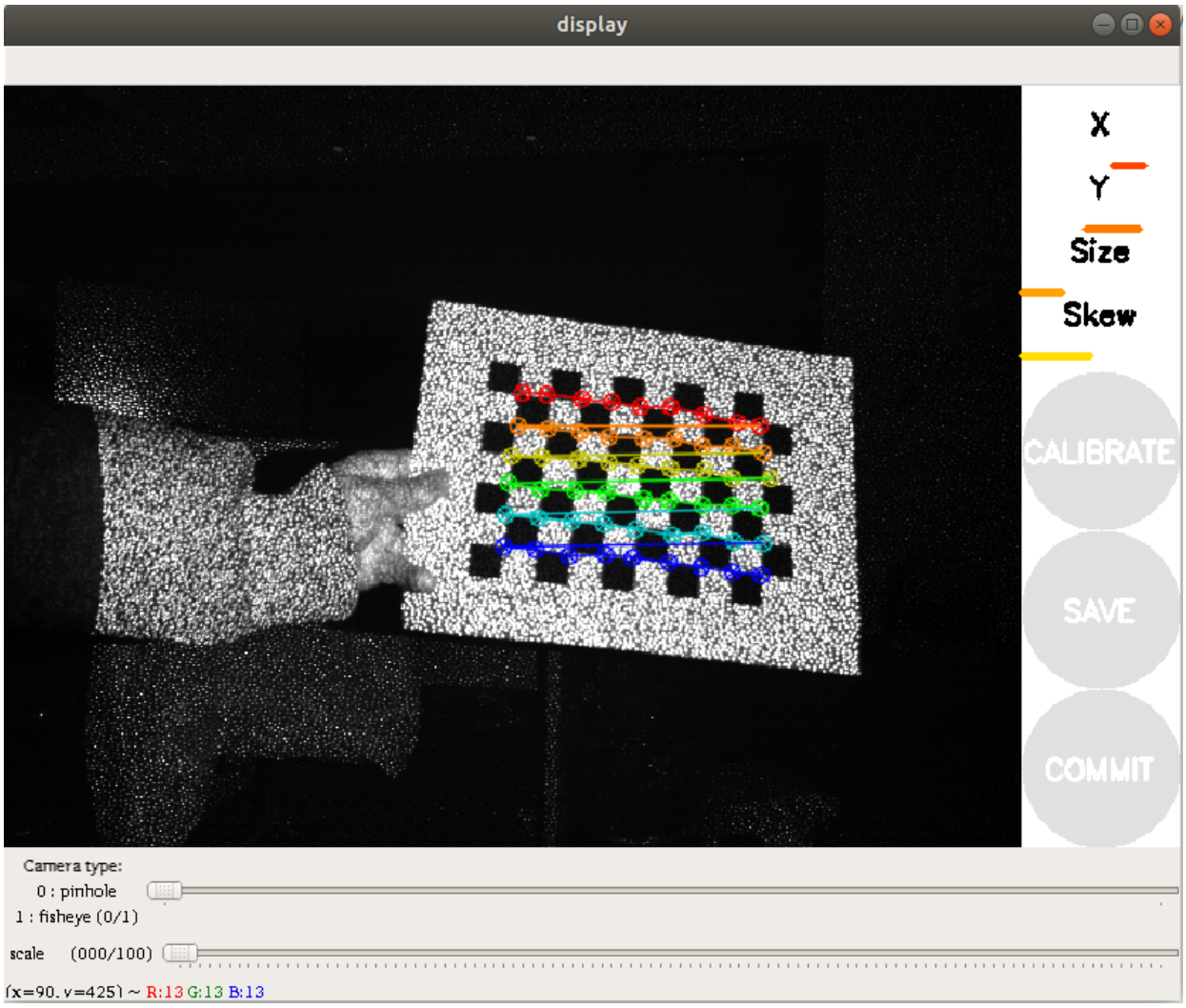
Start the calibration node

```
rosrun camera_calibration cameracalibrator.py image:=/camera/ir/image_mono8  --size
9x6 --square 0.02
```

size: The number of internal corner points of the calibration checkerboard, such as 9X6, the corner points have a total of six rows and nine columns.

square: The side length of the checkerboard, in meters.

image and camera: Set the topic of images posted by the camera.



The following operations are similar to color camera calibration, transforming different poses. The system will recognize it autonomously. In the best case, the lines under [X], [Y], [Size], and [Skew] first turn from red to yellow and then green as the data is collected, filling as much as possible.

- Click [CALIBRATE] to calculate the internal parameters of the camera. The more pictures, the longer the time, just wait. (Sixty or seventy sheets are almost the same, too many are easy to get stuck).
- Click [SAVE] to save the calibration results to [/tmp/calibrationdata.tar.gz].
- Click [COMMIT] to write the calibration file to the [.ros/camera_info/depth_Astra_Orbbec.yaml] file. The calibration result is automatically read the next time the camera is turned on.

After the calibration, you can move out the [/tmp/calibrationdata.tar.gz] file to see the content

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

After decompression, it contains the image just calibrated, an ost.txt file and an ost.yaml file.

2.3 Single target targeting

It is the same as [2.2.1], except that the start command and topic name are different. This subsection is suitable for monocular color image calibration.

Start the monocular camera

```
roslaunch usb_cam usb_cam-test.launch
```

Start the calibration node

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

The single-target calibration results are stored in the [.ros/camera_info/head_camera.yaml] file.