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 internal and external reasons of the camera, the image will be greatly distorted, mainly radial deformation and tangential deformation, causing straight lines to become curved. The farther away from the center of the image, the more serious the distortion. In order to avoid errors caused by data sources, the parameters of the camera need to be calibrated. Calibration essentially uses a known and determined spatial relationship (calibration plate) to reversely deduce the inherent and real parameters of the camera (internal parameters) by analyzing the pixels of the photographed pictures.

Disadvantages of infrared depth camera ranging:

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

(2) It is impossible to accurately measure the distance of specular objects, because only when the depth camera is on the center vertical line of the specular 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 .

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)		

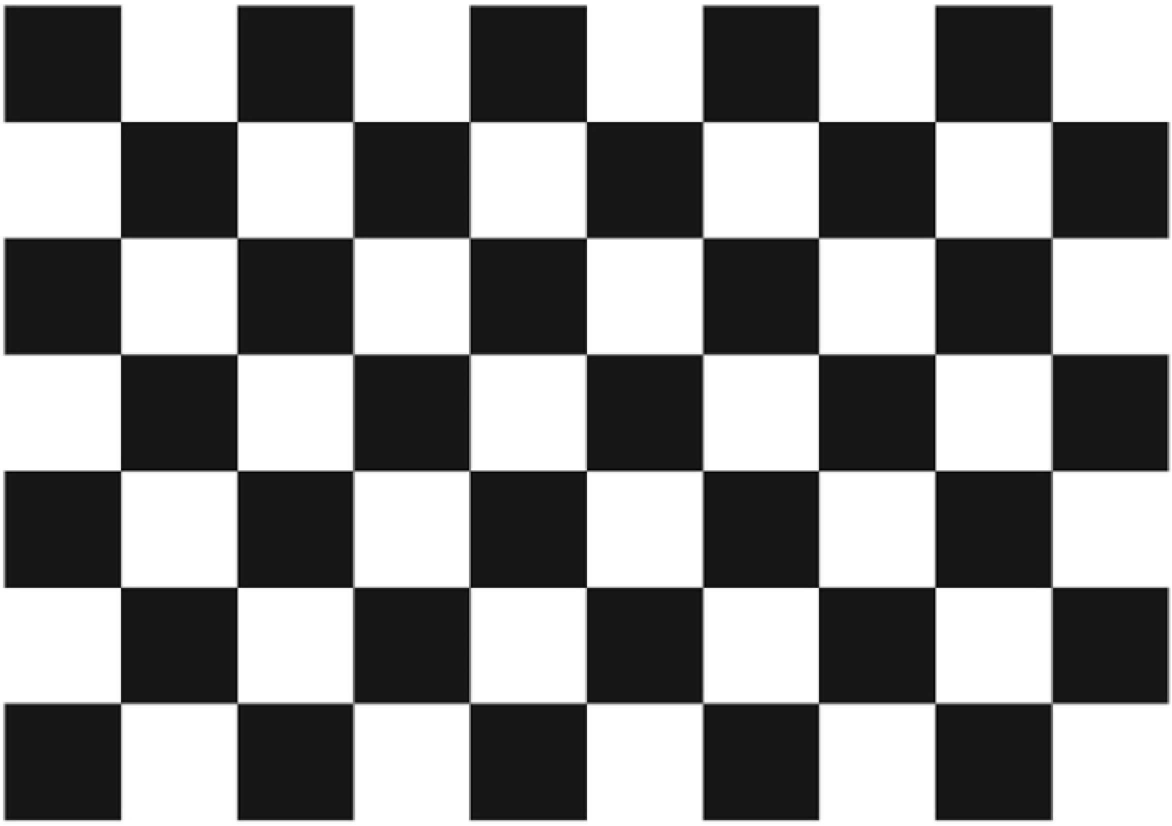
2.1 Preparation before calibration

- A large [chessboard] of known dimensions (http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration?action=AttachFile&do=view&target=check-108.pdf). This tutorial uses a 9x6 checkerboard and a 20mm square, which needs to be flattened when calibrating.

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

  Any calibration board can be used, as long as the parameters are changed.
- An open area without obstacles and calibration board patterns.
- 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
astraproplus.launch	Astra plus/Astraproplus
astrapro.launch	Astra pro
embedded_s.launch	Deeyea
dabai_u3.launch	Dabai
gemini.launch	Gemini

device view

```
lsusb
```

```
jetson@jetson-Yahboom:~$ lsusb
Bus 002 Device 004: ID 2109:0817 VIA Labs, Inc.
Bus 002 Device 003: ID 2109:0817 VIA Labs, Inc. USB3.1 Hub
Bus 002 Device 002: ID 2109:0822 VIA Labs, Inc. USB3.1 Hub
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 006: ID 045e:028e Microsoft Corp. Xbox360 Controller
Bus 001 Device 013: ID 2109:8817 VIA Labs, Inc.
Bus 001 Device 015: ID 2bc5:050f VIA Labs, Inc.          USB2.0 Hub
Bus 001 Device 014: ID 2bc5:060f
Bus 001 Device 012: ID 05e3:0608 Genesys Logic, Inc. Hub
Bus 001 Device 010: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
Bus 001 Device 008: ID 0c45:6340 Microdia Camera
Bus 001 Device 005: ID 2109:2817 VIA Labs, Inc. USB2.0 Hub
Bus 001 Device 011: ID 2109:8817 VIA Labs, Inc.
Bus 001 Device 009: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter
Bus 001 Device 007: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter
Bus 001 Device 004: ID 2109:2817 VIA Labs, Inc.
Bus 001 Device 002: ID 2109:2822 VIA Labs, Inc. USB2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Depth Camera ID: 【2bc5:060f】

Color Camera ID: 【2bc5:050f】

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
```

```
jetson@jetson-Yahboom:~$ rostopic list
/camera/depth/camera_info
/camera/depth/image_raw
/camera/depth/points
/camera/depth_registered/points
/camera/ir/camera_info
/camera/ir/image
/camera/ir/image_mono8
/camera/ir/image_raw
/camera/rgb/camera_info
/camera/rgb/image_raw
/rosout
/rosout_agg
/tf
/tf_static
```

2.2.1 Color icon setting

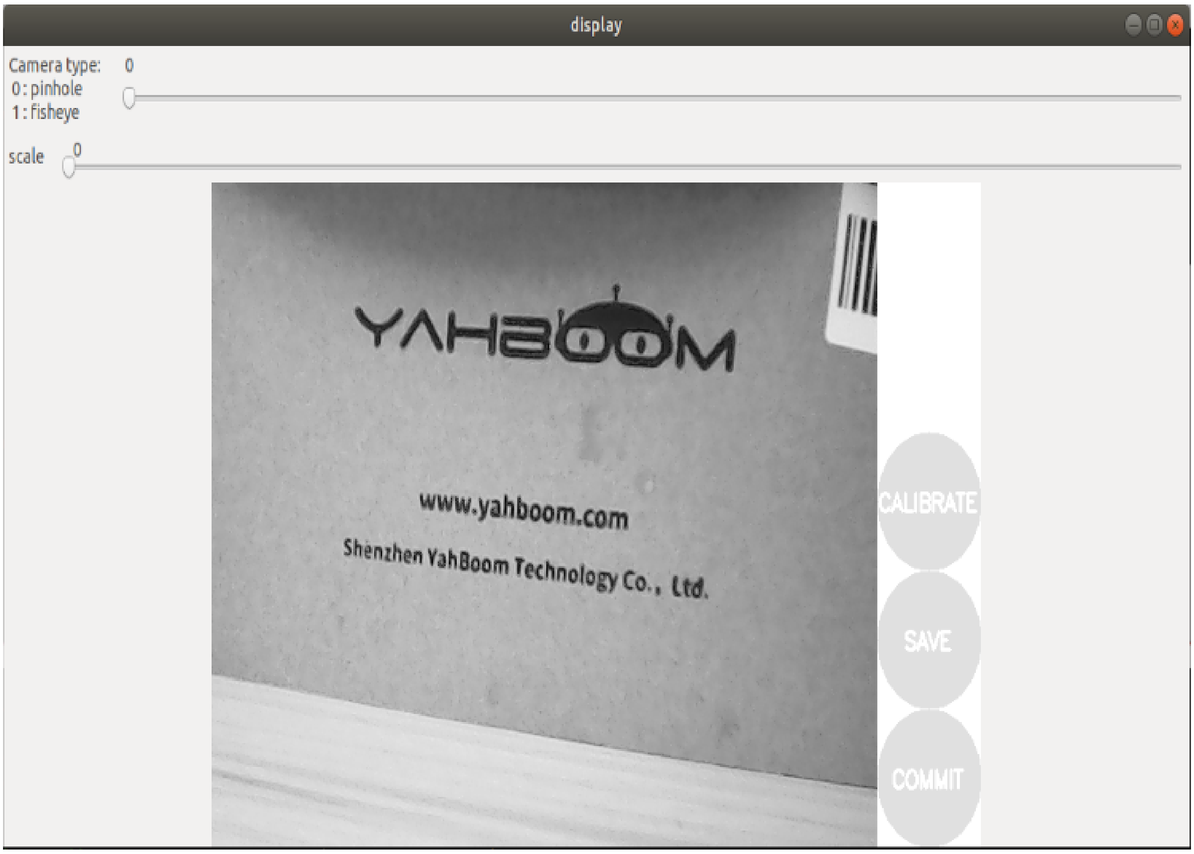
Start calibration node

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

size：Calibrate the number of internal corner points of the checkerboard, for example, 9X6, with a total of six rows and nine columns of corner points.

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

image：Set the image topic released by the camera.



Calibration interface

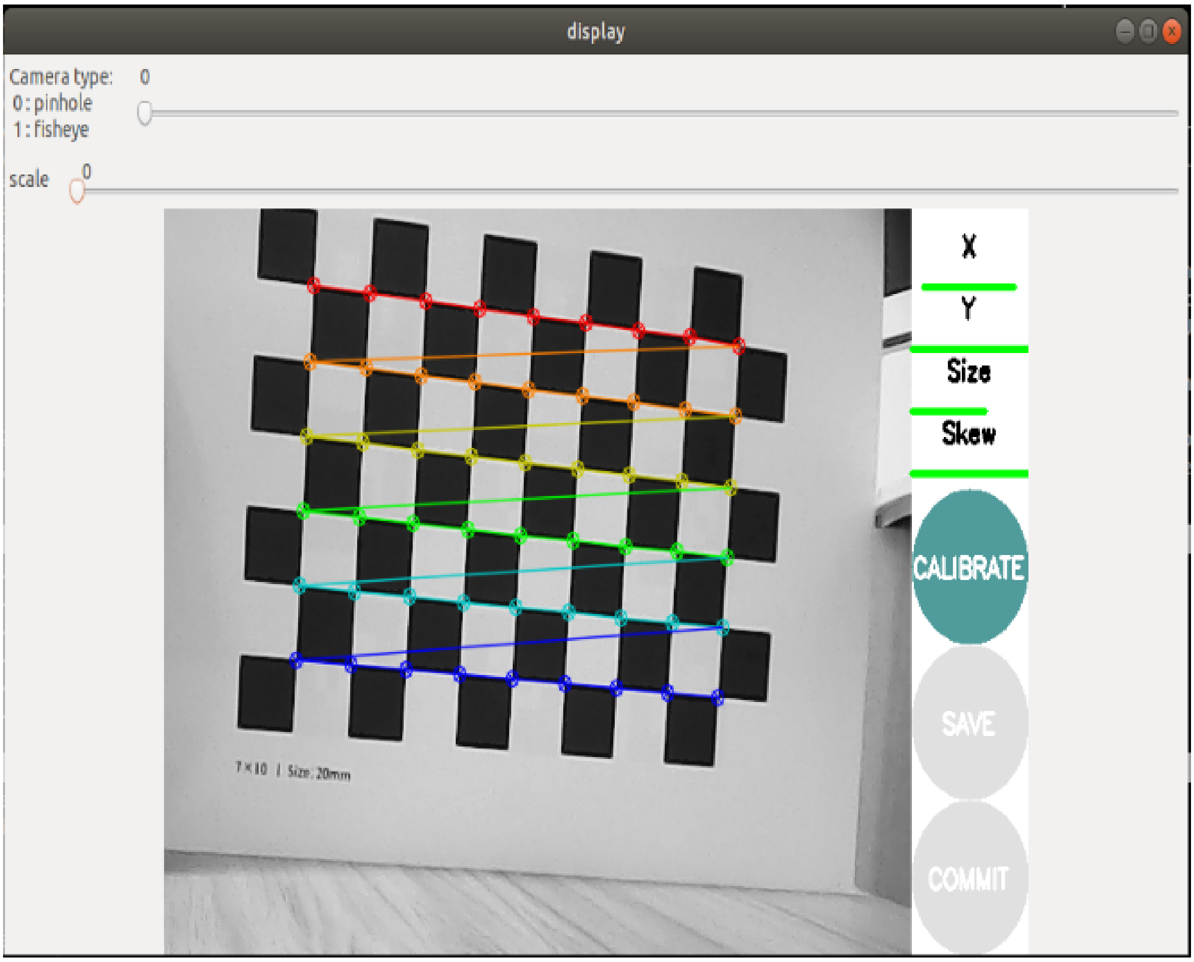
X：The left and right movement of the checkerboard in the camera field of view

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

Size ：The back and forth 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 startup, place the checkerboard in the center of the screen and change to different positions. The system will identify it autonomously. The best situation is that the lines under [X], [Y], [Size], and [Skew] will first change from red to yellow and then to green as the data is collected, filling them as fully as possible.



- Click [CALIBRATE] to calculate the internal parameters of the camera. The more pictures you have, the longer it will take. Just wait. (Sixty or seventy is enough, too many can easily get stuck).
- Click [SAVE] to save the calibration results to [/tmp/calibrationdata.tar.gz].

```
**** Calibrating ****
D = [-0.07028213194362816, 0.00043818252903837866, -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 is completed, you can move out the [/tmp/calibrationdata.tar.gz] file to see the content.

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

After decompression, there are the image just calibrated, an ost.txt file and an ost.yaml file.

2.2.2.IR infrared calibration

After the data normalization problem is dealt with, another problem will arise. Because of the RGBD camera, which uses structured light as the depth imaging principle, the infrared light it projects is a special disordered spot.As a result, the infrared receiving device cannot receive clear and complete picture content.

At this time we can have several special processing methods:

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

Choose the treatment method according to your needs.

Modify the astraproplus.launch file,

```
gedit /home/jetson/software/library_ws/src/orbbec-ros-sdk/launch/astraproplus.launch
```

Change enable_color inside to false, change enable_ir to true, save and exit.

```
    <arg name="connection_delay" default="100" />
    <arg name="color_width" default="640"/>
    <arg name="color_height" default="480"/>
    <arg name="color_fps" default="30"/>
    <arg name="enable_color" default="false"/>
    <arg name="color_format" default="MJPG"/>
    <arg name="flip_color" default="false"/>
    <arg name="enable_color_auto_exposure" default="true"/>
    <arg name="depth_width" default="640"/>
    <arg name="depth_height" default="480"/>
    <arg name="depth_fps" default="30"/>
    <arg name="enable_depth" default="true"/>
    <arg name="depth_format" default="Y11"/>
    <arg name="flip_depth" default="false"/>
    <arg name="ir_width" default="640"/>
    <arg name="ir_height" default="480"/>
    <arg name="ir_fps" default="30"/>
    <arg name="enable_ir" default="true"/>
    <arg name="ir_format" default="Y10"/>
```

Start Astra Camera

```
roslaunch yahboomcar_visual astra_calibration.launch
```

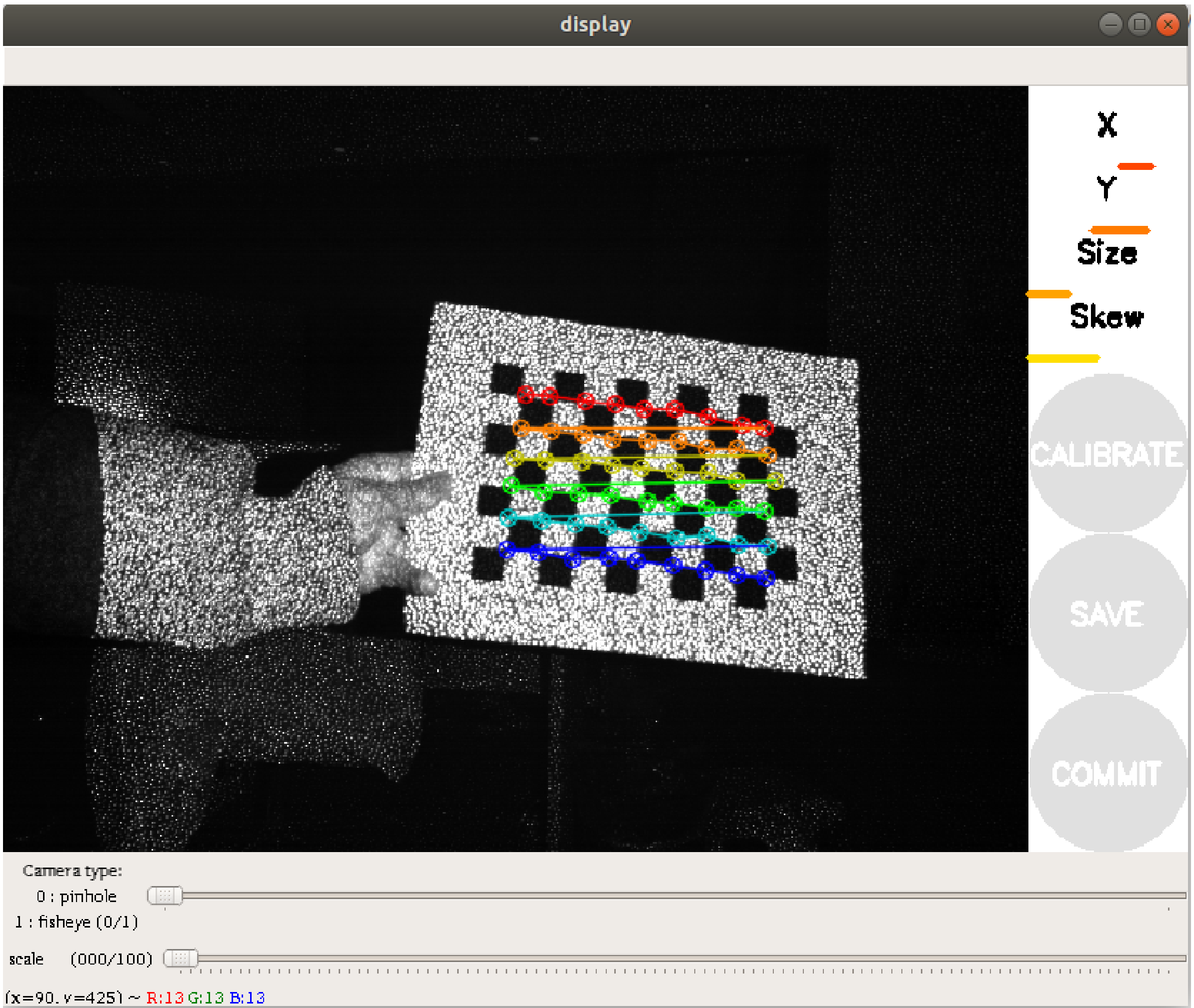
Start calibration node

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

size：Calibrate the number of internal corner points of the checkerboard, for example, 9X6, with a total of six rows and nine columns of corner points.

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

image和camera：Set the image topic published by the camera.



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

- Click [CALIBRATE] to calculate the internal parameters of the camera. The more pictures you have, the longer it will take. Just wait. (Sixty or seventy is enough, too many can easily get stuck).
- Click [SAVE] to save the calibration results 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.

2.3 Single target targeting

The principle of setting the color map in section [2.2.1] is the same, except that the startup command and topic name are different. This section is suitable for calibration of monocular color images.

Start monocular camera

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

Start calibration node

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

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