Camera calibration

1. Preparation before calibration

- A large chessboard of known size. This tutorial uses a 9x6 checkerboard and a 20mm square, which needs to be flattened during calibration** The calibration uses the internal vertices of the chessboard grid, so the "10x7" chessboard uses the internal vertex parameter "9x6", as shown in the following example** Any specification of calibration board is fine, just change the parameters. An open area with no obstacles or calibration board patterns.
- Monocamera for publishing images through ROS

2. Start calibration

Install the calibrated feature pack camera_ calibration, taking noetic as an example.

Input following command:

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

Start the camera before calibration, and then turn off the camera until all calibration is completed.

Input following command to start the camera:

```
#astra camera start up
roslaunch orbbec_camera astra.launch
#gemini2 camera start up
roslaunch orbbec_camera gemini2.launch
```

Input following command to view the topic terminal:

```
rostopic list
```

The topic we need to use to calibrate RGB color images is/camera/color/image_ raw

Input following command to run the calibration program:

```
rosrun camera_calibration cameracalibrator.py image:=/camera/color/image_raw
```

Size: Calibrate the number of internal corner points on the chessboard, such as 9X6, with a total of six rows and nine columns of corner points.

Square: The side length of a chessboard, measured in meters.

Topic name:/camera/color/image_ Raw, if starting **USB_ cam**, modify it to **/image_ raw**
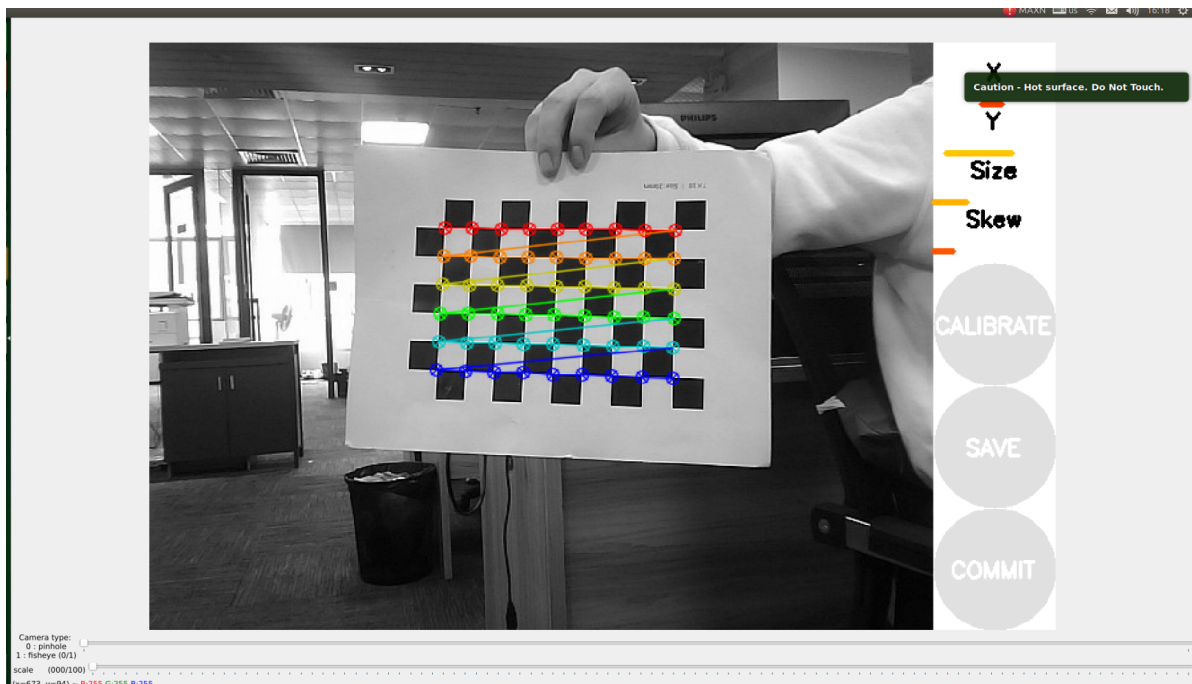
X: The left and right movement of the checkerboard in the camera's field of view

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

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

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

As shown in the above figure, it is necessary to capture the image by flipping it up, down, back, left, right, and left to make the X, Y, Size, and Skew on the right turn green, as shown in the following figure.

Then, click CALIBRATE to start calibration.

After calibration, click SAVE, as shown in the following figure.

Save the calibration results to [/tmp/calibration data. tar. gz], and the saved path is in the terminal directory where the calibration program was started.

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

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

Input following command:

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

We will obtain the calibrated png file, ostyaml, and ost.txt files in the terminal directory.

Due to Dabai_ When the DCW2 driver starts, it loads the built-in parameters calibrated in the code, so there is no need to load the calibrated parameters. However, when starting the USB camera, the parameters need to be loaded.

Therefore, after calibration, it is necessary to replace the original built-in parameters with the parameters and rename the calibrated ost.yaml to camera_ info.yaml, then replace the original camera_ info.yaml

```
sudo mkdir /home/yahboom/.ros/camera_info
sudo cp ost.yaml /home/yahboom/.ros/camera_info
#Switch to/home/yahroom/.ros/camera_info directory
cd /home/yahboom/.ros/camera_info
#Re-name ost.yaml为camera_info.yaml
sudo mv ost.yaml camera_info.yaml
```