4. Target Tracking Algorithm

Locating an object in consecutive frames of a video is called tracking. In OpenCV, you can use traditional target tracking algorithms (such as mean tracking, Kalman filtering, etc.) or deep learning-based target trackers (such as MOSSE, CSRT, etc.) to track targets. Deep learning target trackers usually perform better in accuracy and robustness.

4.1. Use

Source launch file path: /opt/ros/noetic/share/opencv_apps/launch

Step 1: Start the camera

```
roslaunch ascam_visual opencv_apps.launch img_flip:=false
```

- img_flip parameter: whether the image needs to be flipped horizontally, the default is false.

The [usb_cam-test.launch] file opens the [web_video_server] node by default, and you can directly use the [IP:8080] web page to view the image in real time.

Step 2: Start the target tracking function of Opencv_apps

```
roslaunch opencv_apps camshift.launch # Target tracking algorithm
```

Each function case will have a parameter [debug_view], Boolean type, whether to use Opencv to display the image, displayed by default.

If you do not need to display, set it to [False], for example

```
roslaunch opencv_apps camshift.launch debug_view:=False
```

However, after starting in this way, some cases may not be displayed in other ways, because in the source code, some [debug_view] is set to [False], which will turn off the image processing.

4.2, Display method

- rqt_image_view

Enter the following command and select the corresponding topic

```
rqt_image_view
```

- opencv

The system displays by default, no processing is required.

- Web page viewing

(Under the same LAN) Enter IP+port in the browser, for example:

```
192.168.2.116:8080
```

For specific IP, use your current virtual machine IP.

4.3, Effect display

You can see an adjustable window appear on the screen, and a red frame appears to follow.