3、HD camera object tracking

3.1、Introduction

Website：https://learnopencv.com/object-tracking-using-opencv-cpp-python/#opencv-tracking-api

Object tracking is to locate an object in consecutive video frames.

- Comparison of OpenCV algorithms

Algorithm	Speed	Accuracy	Description
BOOSTING	Slow	Low	It is the same as the machine learning algorithm behind Haar casades (AdaBoost), but it has been born for more than ten years, a veteran algorithm.
MIL	Slow	Low	It is more accurate than BOOSTING, but the failure rate is higher.
KCF	Fast	High	Faster than BOOSTING and MIL, but it is not effective when there is occlusion
TLD	Middle	Middle	There are a lot of erro
MEDIANFLOW	Middle+	Middle	The model will fail for fast-jumping or fast-moving objects.
GOTURN	Middle	Middle	A deep learning-based object detector requires additional models to run.
MOSSE	Fastest	High	The speed is really fast, but not as high as the accuracy of CSRT and KCF. If you are looking for speed, you can choose it.
CSRT	Fast -	Higher	Slightly more accurate than KCF, but not as fast as KCF.

3.2、Steps

Note: The [R2] of the handle remote controller can [Pause/Open] for all functions of robot car

3.2.1、Start up

Start the bottom driver control, and it can also be placed in other launch files. (Jetson nano side)

```
roslaunch transbot_bringup bringup.launch
```

Method 1

Start up HD camera（Raspberry Pi side）

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

Start HS camera target tracking control（virtual machine）

```
roslaunch transbot_mono mono_tracker.launch VideoSwitch:=false tracker_type:=KCF
```

Method 2

Note: press【q】key to exit.

```
python3 ~/transbot_ws/src/transbot_mono/scripts/mono_Tracker.py
```

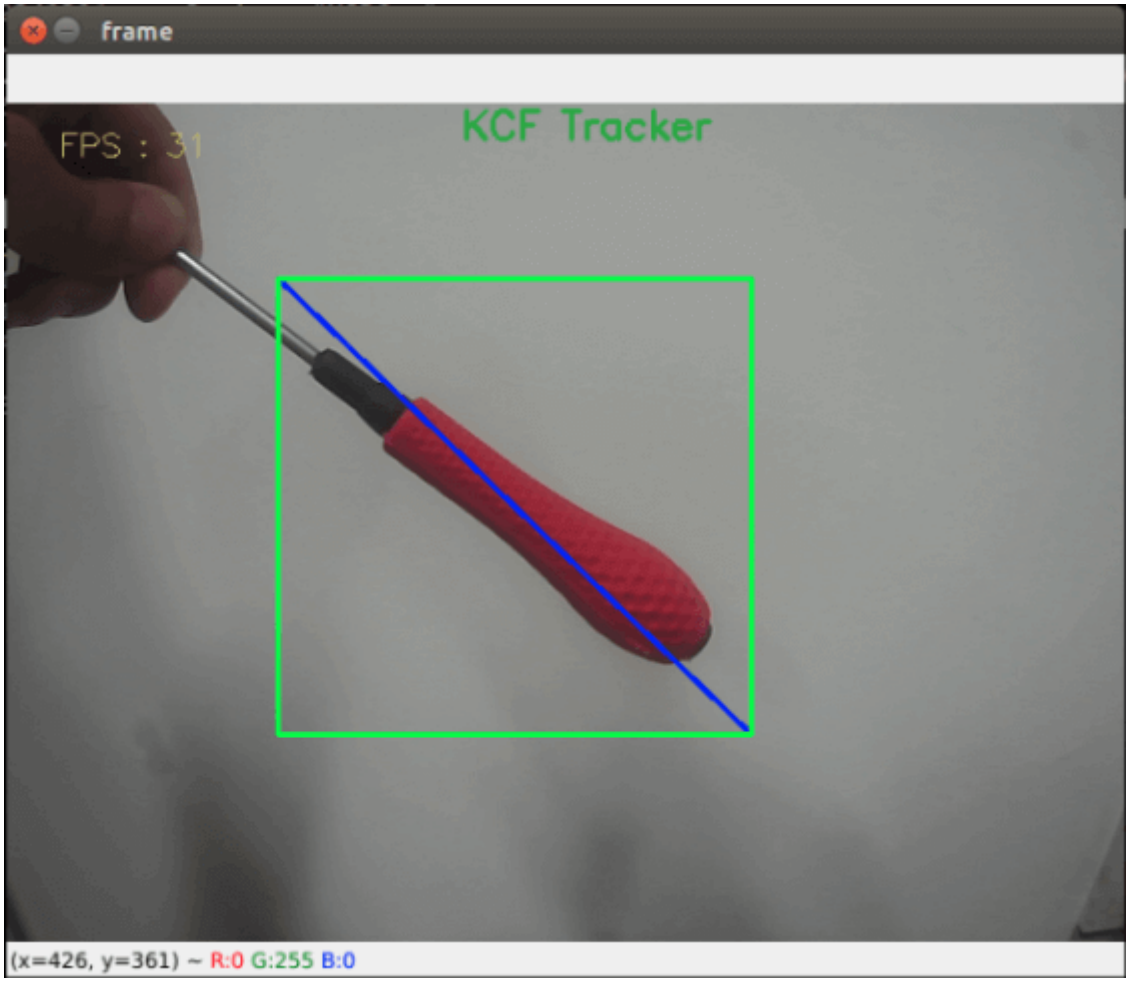
This method can only be activated in the master controller that the camera is connected.

- VideoSwitch parameter: whether to use the camera function package to start; for example: start usb_cam-test.launch, this parameter must be set to true; otherwise, it is false.
- tracker_type parameter: OpenCV Tracking API; optional: ['BOOSTING','MIL','KCF','TLD','MEDIANFLOW','MOSSE','CSRT']

Set the parameters according to your needs, and you can also modify the launch file directly, so you don't need to attach parameters when you start.

3.2.2、Identify

After starting, enter the selection mode, use the mouse to select the location of the object, as shown in the figure below, release it to start recognize.



Keyboard key control:

【r】： Color selection mode, the mouse can be used to select the area of the color to be recognized (cannot exceed the area range).

【f】: Switching algorithm: ['BOOSTING','MIL','KCF','TLD','MEDIANFLOW','MOSSE','CSRT','color'].

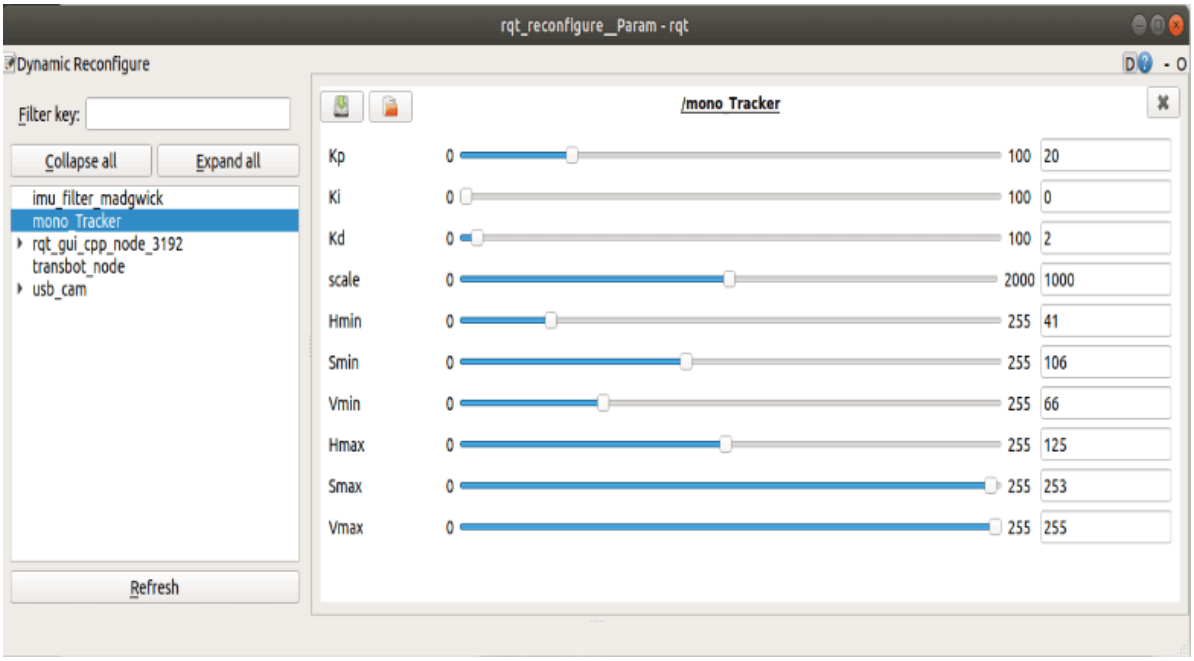
【q】: Exit the program.

【Space key】: Color follow. When following, we need to move the target slowly, moving too fast it will lose the target.

3.2.3、PID adjustment

Dynamic parameter

```
rosrun rqt_reconfigure rqt_reconfigure
```



Select [mono_Tracker] node, [Hmin], [Smin], [Vmin], [Hmax], [Smax], [Vmax] these six parameters do not need to be adjusted.
When the slider is in the dragging state, no data will be transferred to the system.
When we release the slider, the data will be transferred to the system; we can also select a row and then slide the mouse wheel.

Parameter analysis:

[Kp], [Ki], [Kd]: PID control during the movement of the robot car.

[Scale]: PID scaling.

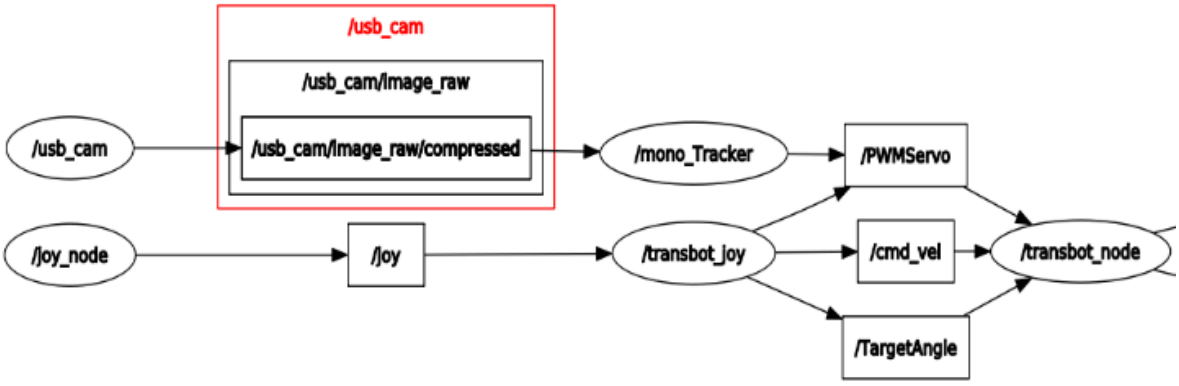
3.2.4、Target follow

After identifying is ok, click [Space key] on the keyboard to execute the color following program.

- View node

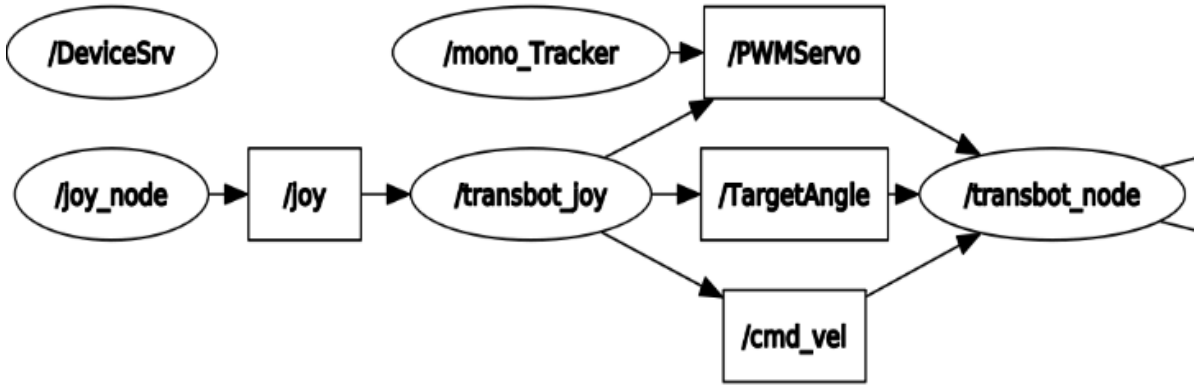
```
rqt_graph
```

- Method1 start up，node【mono_Tracker】



Subscribe to image topics; publish gimbal servo topics

- Method2--start up，node【mono_Tracker】



Subscribe to image topics; control gimbal servo following target object.