5、KCF object tracking

5.1、Introduction

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

- Transbot Se robot KCF target tracking, can track the target object at any time, and follow the target object to move on the table, transbot se (left and right) and mechanical arm (up and down) follow the target at the same time
- Object Tracking

  Object tracking is to locate an object in consecutive video frames. This definition sounds straightforward, but in computer vision and machine learning, tracking is a very broad term that encompasses concepts that are conceptually similar but technically different. For example, all of the following different but related ideas are commonly studied under object tracking:

  （1）（Dense Optical flow DOF）：These algorithms help to estimate the motion vector of each pixel in a video frame.

  （2）（Sparse optical flow）:An example is the Kanade-Lucas-Tomashi (KLT) feature tracking algorithm, which tracks the position of several feature points in an image.

  （3）（Kalman Filtering）:A very popular signal processing algorithm based on prior motion information is used to predict the position of a moving object. One of the early applications of this algorithm was missile guidance! The onboard computer that guided the Apollo 11 lunar module to the moon had a Kalman filter.

  （4）Meanshift and Camshift：This is the algorithm that locates the maximum of the density function and they are also used for tracking.

  （5）（Single object trackers）:In this class of trackers, the first frame uses a rectangular marker to indicate the location of the object to be tracked. The object is then tracked in subsequent frames using a tracking algorithm. In most real-world applications, these trackers are used together with object detectors.

  （6）（Multiple object track finding algorithms）:When we have a fast object detector, it makes sense to detect multiple objects in each frame and then run a trace finding algorithm to identify which rectangle in one frame corresponds to the rectangle in the next frame.

- OpenCV Tracking API

  OpenCV 4 comes with a tracking API that contains implementations of many single-object tracking algorithms.

OpenCV 4.2 implements these eight trackers -Boosting, MIL, KCF, TLD, MEDIANFLOW, GOTURN, MOSSE, and CSRT.

OpenCV 3.2 implements the six trackers -Boosting, MIL, TLD, MEDIANFLOW, MOSSE, and GOTURN.

OpenCV 3.1 implements these five trackers -Boosting, MIL, KCF, TLD, MEDIANFLOW.

- 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.

5.2、Steps

Note: The [R2] of the handle remote controller can [Pause/Open] for all functions of robot car.Before using this function, you need to adjust the value of [mindist] to 0.

3.2.1、Start up

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

```
roslaunch transbot_track TrackSrv.launch
```

Method 1

Start up HD camera（robot side）

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

Start HD camera target tracking control（virtual machine）

```
roslaunch transbot_track Tracker.launch VideoSwitch:=false tracker_type:=KCF
```

Method 2

Note: press【q】key to exit.

Start monocular target tracking control (robot side)

```
roslaunch transbot_track Tracker.launch VideoSwitch:=true tracker_type:=KCF
```

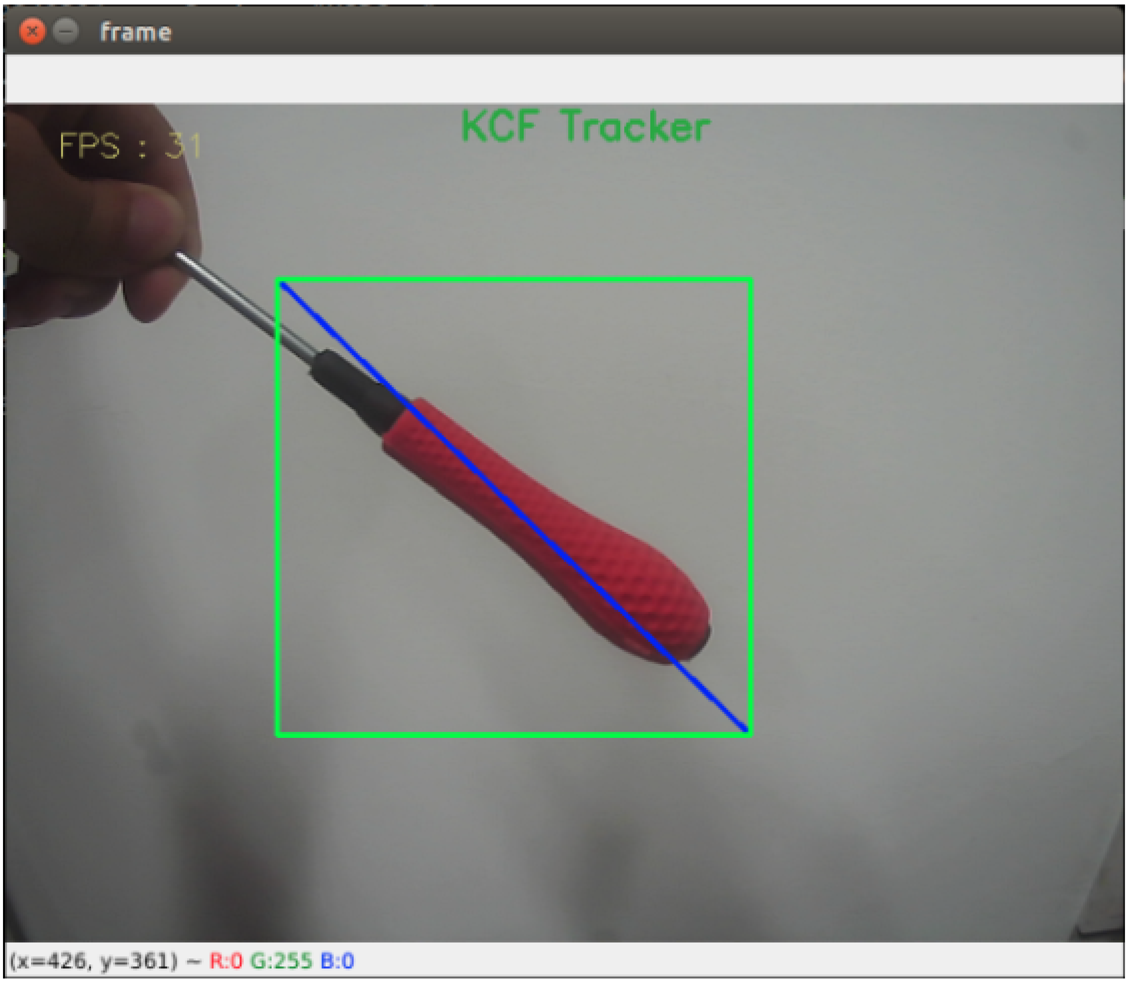
This method can only be activated in the master controller that the camera is connected.

- 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.

5.2.2、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.



(x=426, y=361) ~ R:0 G:255 B:0

Keyboard key control:

【r】: Select mode, you can use the mouse to select the area to identify the target, as shown in the figure above. If the robot arm blocks the camera, press the 【r】key to reset the robot arm.

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

【q】: Exit the program.

【Space key】: Object tracking; Move the target slowly when following, move too fast and you will lose the target.

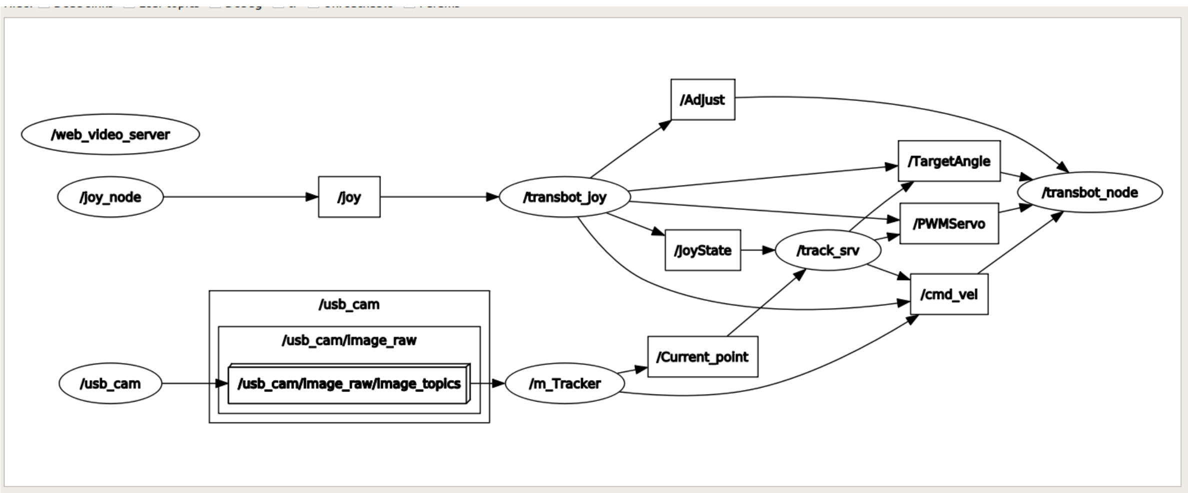
5.2.3、Target Tracking

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

- View node

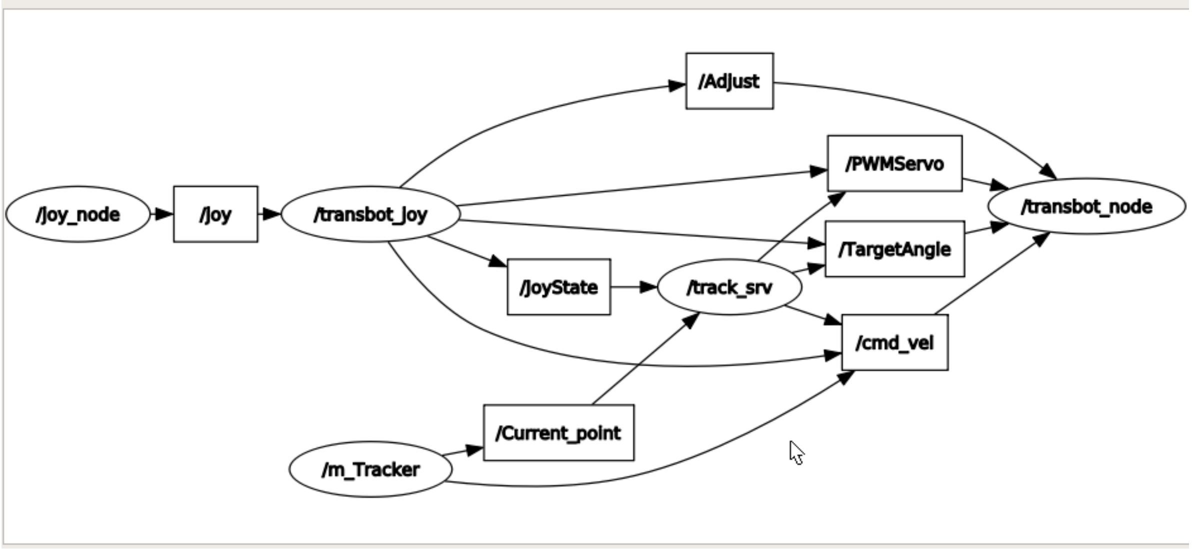
```
rqt_graph
```

- Method1 start up，node【m_Tracker】



Subscribe to image topics; publish gimbal servos, robotic arms, and chassis drive topics

- Method2--start up，node【m_Tracker】



Publish the topics of gimbal servos, chassis drivers and robotic arms.