8、Color recognition and object tracking

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

8.1、website

8.1.1、Introduction

The color recognition function can recognize multiple colors at any time, and independently store the currently recognized color, and detect the color in real time. It can also realize the function of HSV real-time regulation. By adjusting the high and low thresholds of HSV, the interfering colors can be filtered out, so that color blocks can be ideally recognized in complex environments.

- **HSV**

The parameters of the color in this model are: hue (H), saturation (S), and lightness (V).

H: 0 — 180

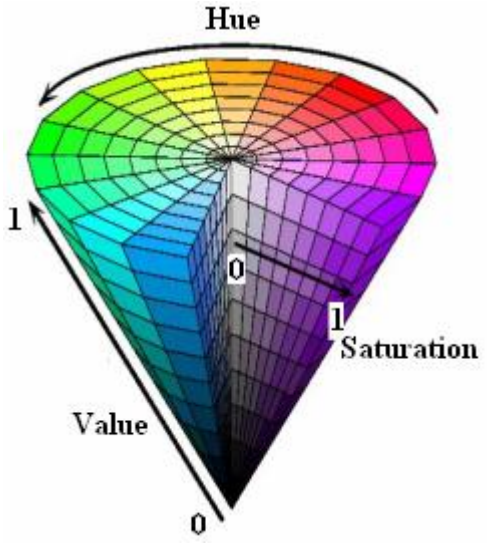
S: 0 — 255

V: 0 — 255

Part of the red is classified as the purple range here:

	black	grey	white	red		orange	yellow	green	light blue	blue	Purple
H_min	0	0	0	0	156	11	26	35	78	100	125
H_max	180	180	180	10	180	25	34	77	99	124	155
S_min	0	0	0	43		43	43	43	43	43	43
S_max	255	43	30	255		255	255	255	255	255	255
V_min	0	46	221	46		46	46	46	46	46	46
V_max	46	220	255	255		255	255	255	255	255	255

-



8.1.2、Object tracking

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.

8.2、Steps

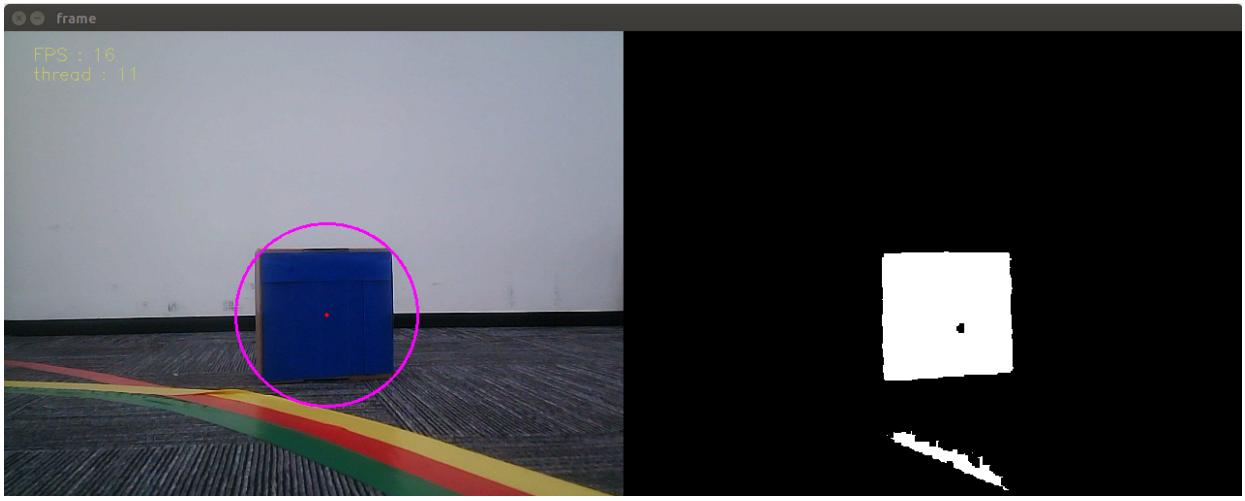
8.2.1、Start up

- color tracking

```
roslaunch astra_tracker AstraTracker.launch
```

8.2.2、Identify

After startup, the system defaults to **[Target Detection Mode]**, as shown below.



Keyboard key control:

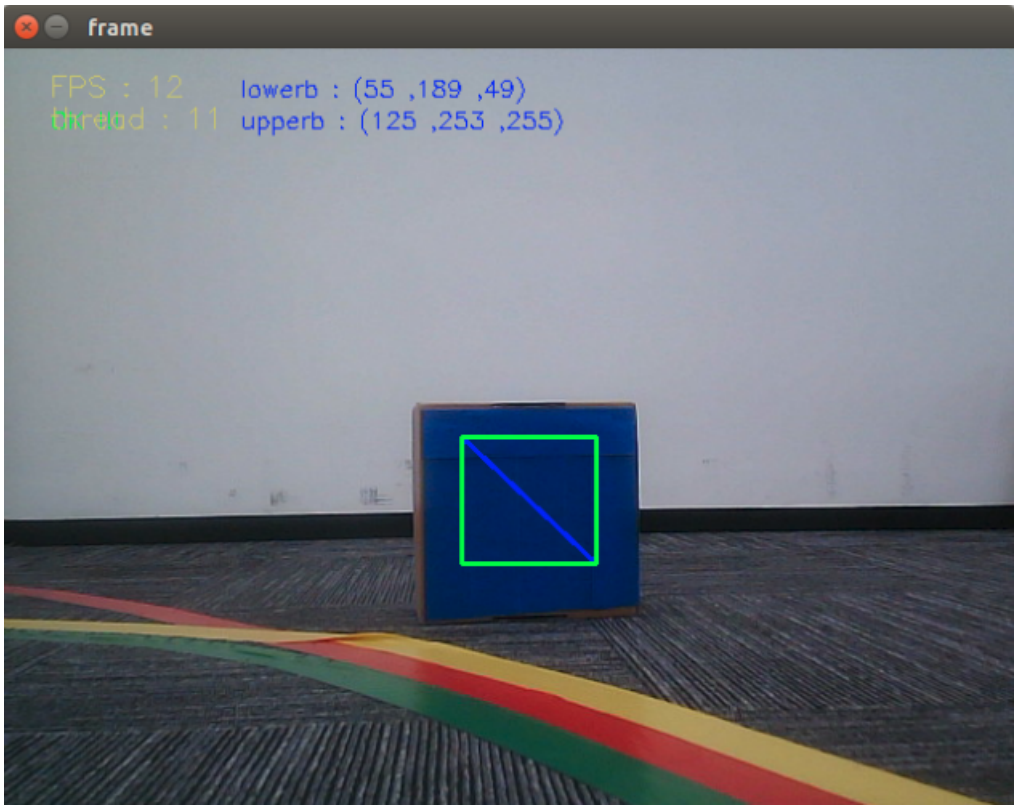
【r】： Color selection mode, the mouse can be used to select the area of the color to be recognized (cannot exceed the area range).

【i】: Target detection mode. Color map on the left (Color), binary map on the right (Binary).

【q】: Exit the program.

- Color recognition

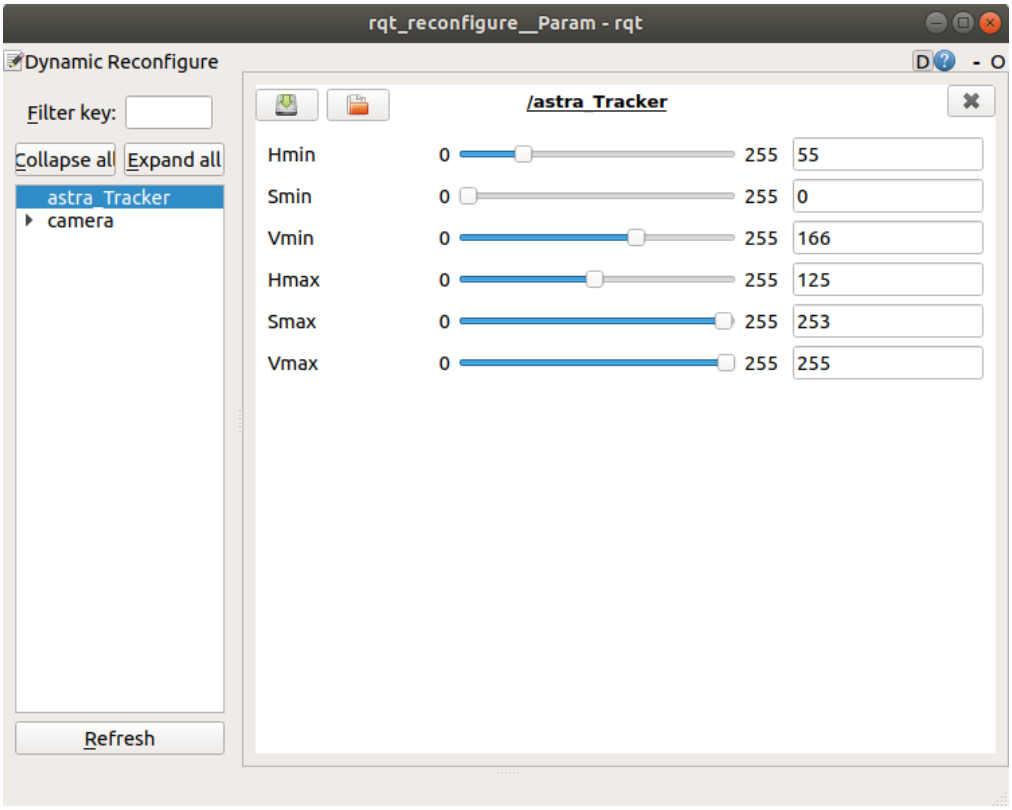
In the color selection mode, use the mouse to select the location of the colored object, as shown in the figure below, release it to start recognition.



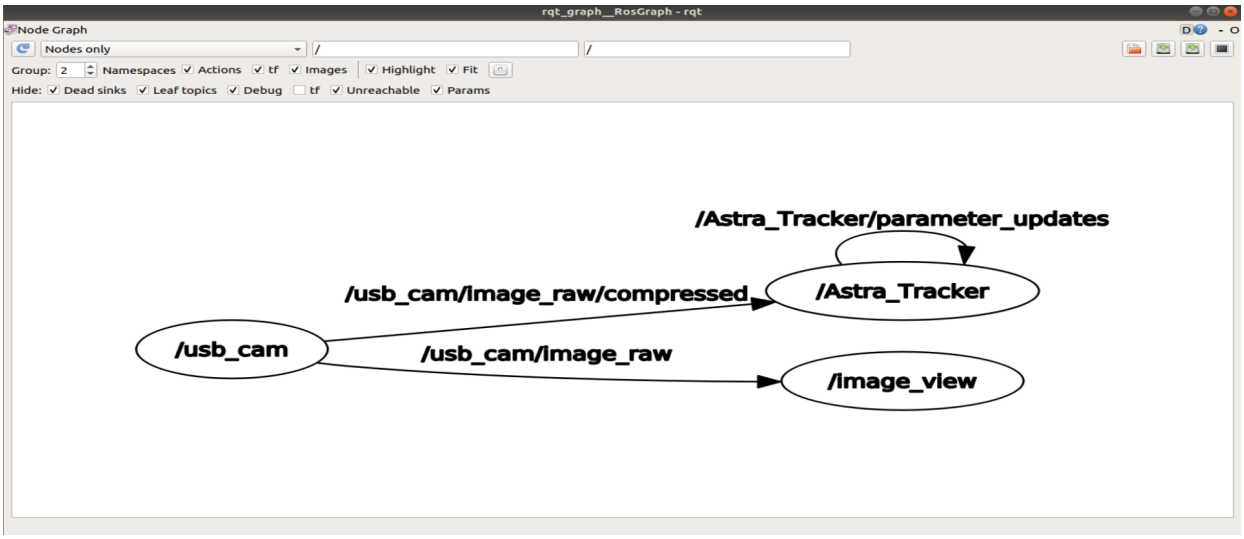
8.2.3、Object tracking

Dynamic parameter tuning

```
rosrun rqt_reconfigure rqt_reconfigure
```



Select the [astra_Tracker] node, generally only need to adjust **[Hmin], [Smin], [Vmin], [Hmax],** these four parameters can be well identified. The slide bar is always in the dragging state, and the data will not be transmitted to the system, it can only be released after it is released; you can also select a row, and then slide the mouse wheel.



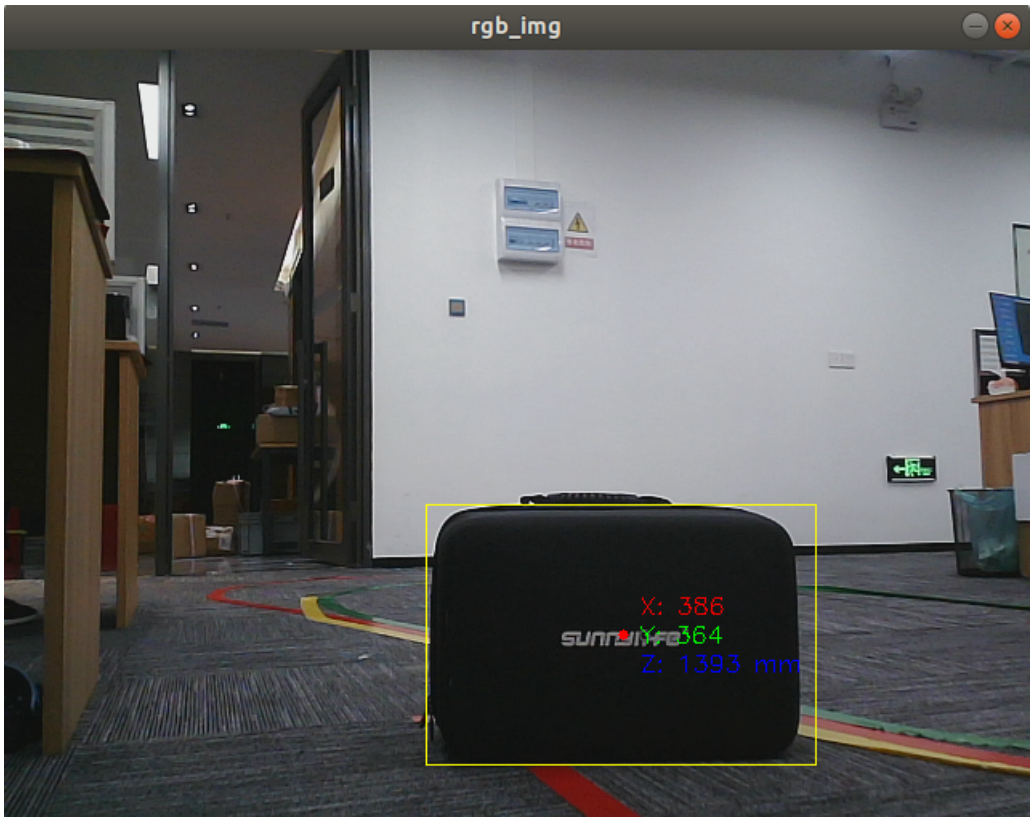
8.3、C++ version KCF

8.3.1、Instructions

Start up

```
roslaunch astra_tracker KCFTracker.launch
```

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 recognition.



8.3.3、Node relationship graph

```
rqt_graph
```