5. Visual tracking autopilot

Feature pack: ~/yahboomcar_ws/src/yahboomcar_linefollw

5.1 Introduction

The Yahboom mobile robot depth camera drives autonomously. It can recognize multiple colors at any time, store the currently recognized color independently, and follow the detected color. During the tracking process, it can also realize the function of real-time obstacle avoidance.

The Yahboom mobile robot can also realize the function of real-time HSV control. By adjusting the high and low thresholds of HSV, the color of interference is filtered out, so that the tracking route can be ideally identified in complex environments. If so, at this time, we need to move the car to a different environment to calibrate it so that we can recognize the color we need in a complex environment.

5.1.1 Introduction to HSV

HSV(Hue, Saturation, Value) is a color space created by AR Smith in 1978 based on the intuitive characteristics of color, also known as the Hexcone Model.

The parameters of the color in this model are: Hue(H), Saturation(S), Lightness(V).

H: 0 -- 180

S: 0 -- 255

V: 0 -- 255

Part of the red is classified as purple 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

5.1.2 HSV hexagonal pyramid

- **Hue H**

Represents color information, that is, the position of the spectral color at which it is located. This parameter is represented by an angle, and the value range is 0° ~ 360°. It is calculated counterclockwise from red, red is 0°, green is 120°, and blue is 240°. Their complementary colors are: yellow at 60°, cyan at 180°, and purple at 300°.

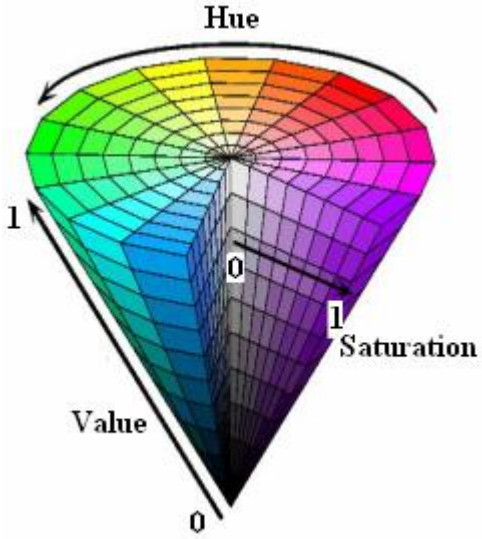
- **Saturation S**

Saturation S is expressed as the ratio between the purity of the selected color and the maximum purity of that color. When S=0, there is only grayscale. 120 degrees apart. Complementary colors are 180 degrees apart. A color that can be seen as the result of a spectral color mixed with white. The greater the proportion of spectral colors, the higher the degree of color close to spectral colors, the higher the color saturation. The saturation is high, and the color is deep and vivid. The white light component of the spectral color is 0, and the saturation is the highest. Usually the value ranges from 0% to 100%, the larger the value, the more saturated the color.

- **Brightness V**

Lightness indicates the brightness of the color. For light source color, the lightness value is related to the brightness of the illuminant; for object color, this value is related to the transmittance or reflectance of the object. Usually the value ranges from 0%(black) to 100%(white). One thing to note: there is no direct connection between it and light intensity.

The 3D representation of the HSV model evolves from an RGB cube. Imagine looking from the white vertices to the black vertices along the diagonal of the cube, and you can see the hexagonal shape of the cube. The hexagonal borders represent color, the horizontal axis represents purity, and lightness is measured along the vertical axis.



5.2 Operation steps

5.2.1 Start

Note: [R2] of the remote controller has the function of [pause/on] for this gameplay. Different models, the parameter range will change, the principle is the same; take the X3 wheat wheel car as an example.

Two startup methods, choose one, and the demo case is method 2; place the robot at the starting position before starting, so that the depth camera faces down as much as possible

method one

robot side

```
roslaunch  yahboomcar_linefollw  follow_line.launch  VideoSwitch:=true
img_flip:=false
```

Method 2

It can be controlled remotely for easy operation.

robot side

```
roslaunch  yahboomcar_linefollw  follow_line.launch  VideoSwitch:=false
img_flip:=false
```

virtual machine

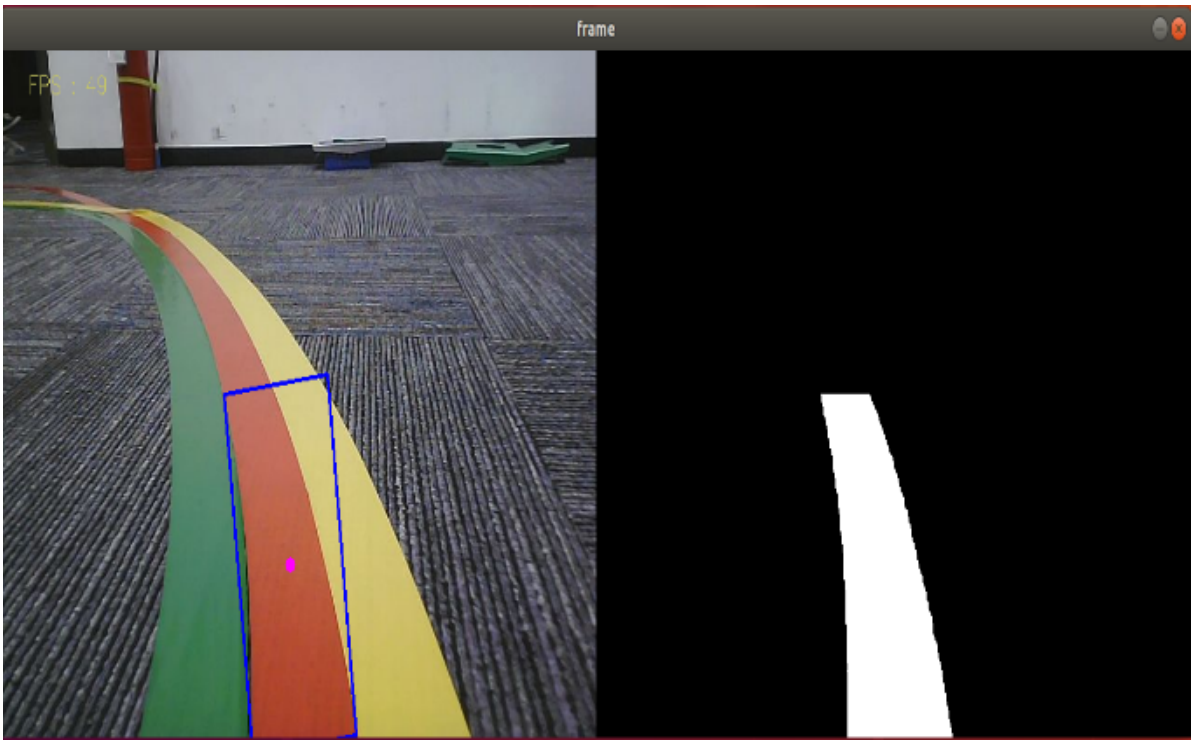
```
roslaunch yahboomcar_linefollw line.launch
```

- VideoSwitch parameter: whether to use the camera function package to start.
- img_flip parameter: whether to flip the screen horizontally, the default is false.

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

5.2.2 Identification

After the startup is completed, the system defaults to [target detection mode], as shown in the lower left figure:



Keyboard key control:

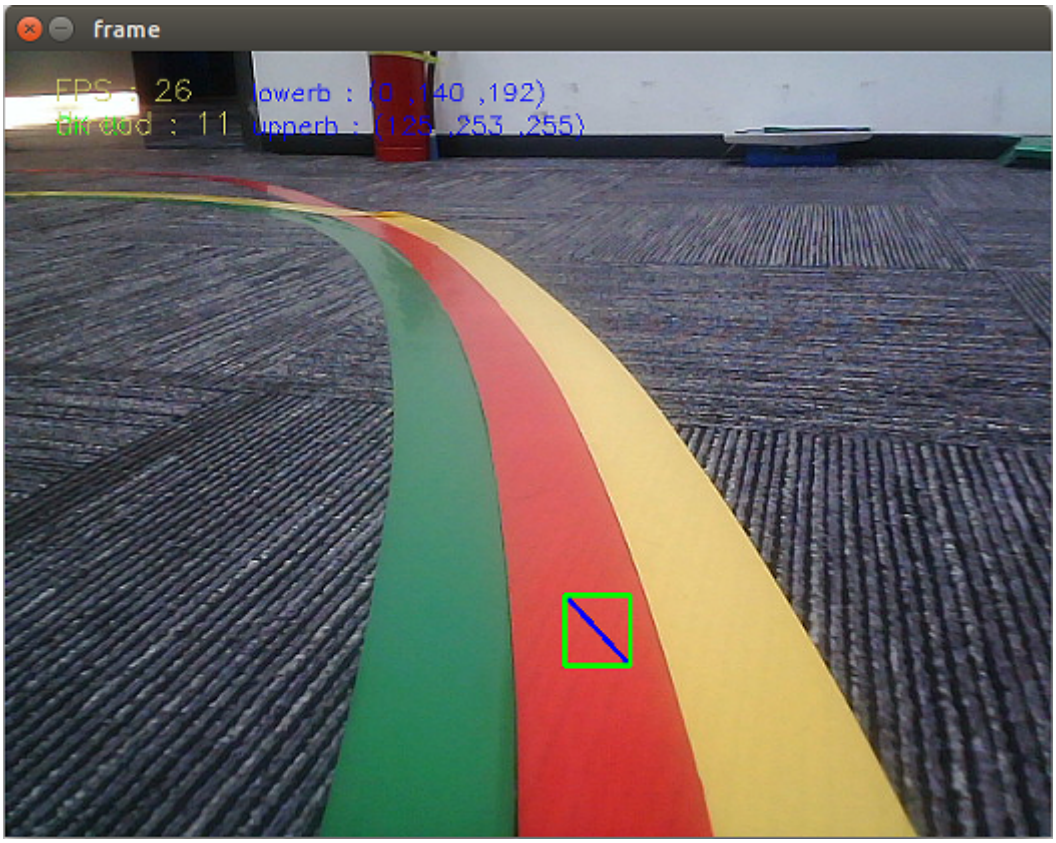
[r]: Color selection mode, you can use the mouse to select the area of the color to be recognized(cannot exceed the area range).

[i]: Target detection mode. Color image on the left(Color), and binary image on the right(Binary).

[q]: Exit the program.

[Spacebar]: Follow the track.

In the color selection mode, you can use the mouse to select the area of the color to be recognized(can not exceed the area range), as shown in the figure below, release it to start the recognition.

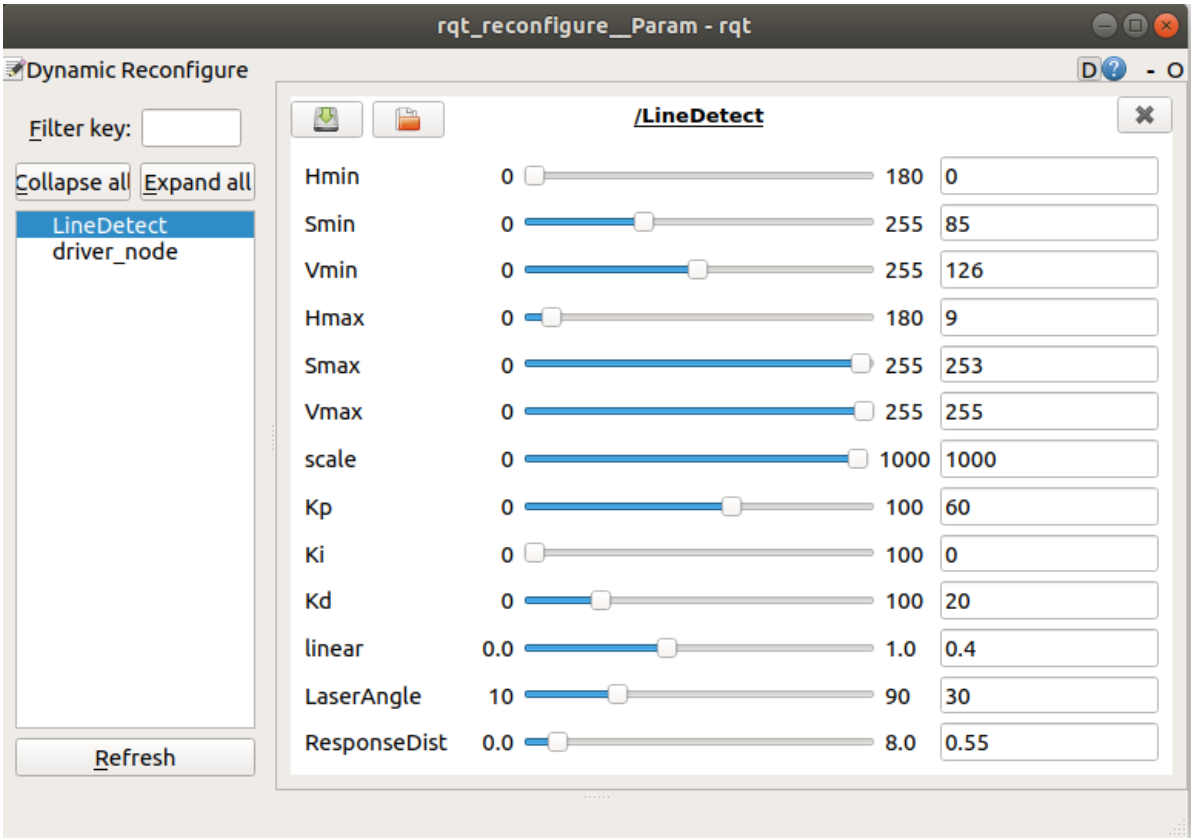


5.2.3 Color calibration

Dynamic parameter debugging tool

```
rosrun rqt_reconfigure rqt_reconfigure
```

Set the mode to [target detection mode] and start the dynamic parameter debugging tool.



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

Parameter parsing:

[Kp], [Ki], [Kd]: PID control during the driving process of the car.

[scale]: PID scaling.

[linear]: The running speed of the trolley; range [0, 1.0], unit: m; set as required.

[LaserAngle]: Effective angle of lidar; range [0, 180], unit: degree; set according to requirements.

[ResponseDist]: Lidar response distance; range [0.15, 8.0], unit: m; set as required.

- parameter modification

When the parameters are adjusted to the optimal state, modify the corresponding parameters to the file, and there is no need to adjust them when using them again.

According to the optimal parameters of the [rqt_reconfigure] debugging tool, enter the [scripts] folder of the [yahboomcar_linefollw] function package, and modify the parameters corresponding to the [follow_line.py] file, as shown below

```
class  LineDetect :
     def __init__(self):
        rospy.on_shutdown(self.cancel)
        rospy.init_node("LineDetect", anonymous = False)
 …
        self.scale  =  1000
        self.FollowLinePID  =(60, 0, 20)
        self.linear  =  0.4
        self.LaserAngle  =  30
        self.ResponseDist  =  0.55
        self.PID_init()
 …
```

[rqt_reconfigure] Modify the initial value of the debugging tool

```
gen.add("Hmin", int_t, 0, "Hmin in HSV", 0, 0, 180)
gen.add("Smin", int_t, 0, "Smin in HSV", 85, 0, 255)
gen.add("Vmin", int_t, 0, "Vmin in HSV", 126, 0, 255)
gen.add("Hmax", int_t, 0, "Hmax in HSV", 9, 0, 180)
gen.add("Smax", int_t, 0, "Smax in HSV", 253, 0, 255)
gen.add("Vmax", int_t, 0, "Vmax in HSV", 255, 0, 255)
gen.add("scale", int_t, 0, "scale", 1000, 0, 1000)
gen.add("Kp", int_t, 0, "Kp in PID", 60, 0, 100)
gen.add("Ki", int_t, 0, "Ki in PID", 0, 0, 100)
gen.add("Kd", int_t, 0, "Kd in PID", 20, 0, 100)
gen.add("linear", double_t, 0, "linear", 0.4, 0, 1.0)
gen.add("LaserAngle", int_t, 0, "LaserAngle", 30, 10, 90)
gen.add("ResponseDist", double_t, 0, "ResponseDist", 0.55, 0, 8)
exit(gen.generate(PACKAGE, "LineDetect", "LineDetectPID"))
```

Enter the [cfg] folder of the [yahboomcar_linefollw] function package, and modify the initial values of the parameters corresponding to the [LineDetectPID.cfg] file. The color [HSV] adjustment parameters can be left unchanged, and the system will automatically generate the [LineFollowHSV.text] file, which will be automatically read when the system starts.

```
gen.add("Kp", int_t, 0, "Kp in PID", 60, 0, 100)
```

Analysis of the above one as an example

parameter	Parse	Corresponding parameters
name	the name of the parameter	"Kp"
type	parameter data type	int_t
level	a bitmask passed to the callback	0
description	a description parameter	"Kp in PID"
default	Initial value for node startup	60
min	parameter minimum	0
max	parameter maximum	10.0

Note: After modification, the update environment must be recompiled to be effective.

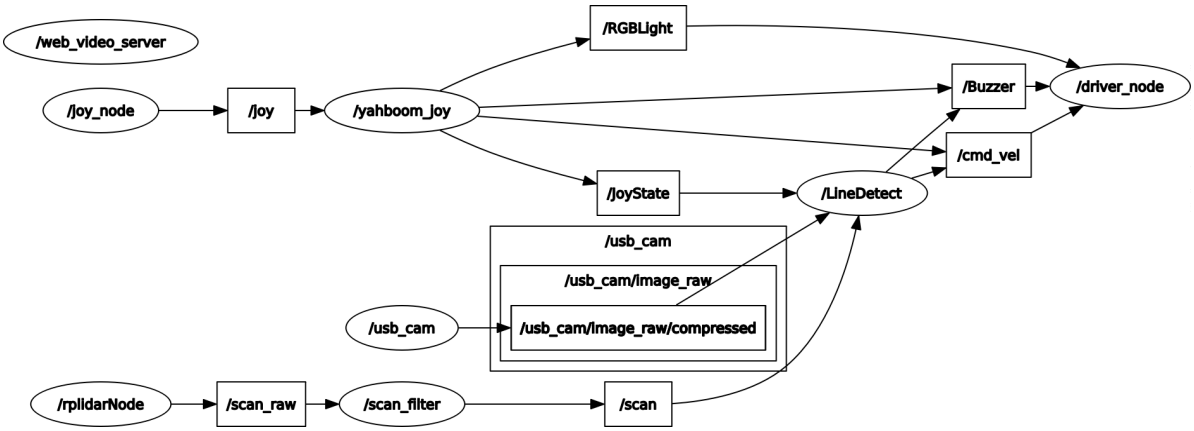
```
cd  ~/ yahboomcar_ws
catkin_make
source  devel/setup.bash
```

5.2.4 Track driving

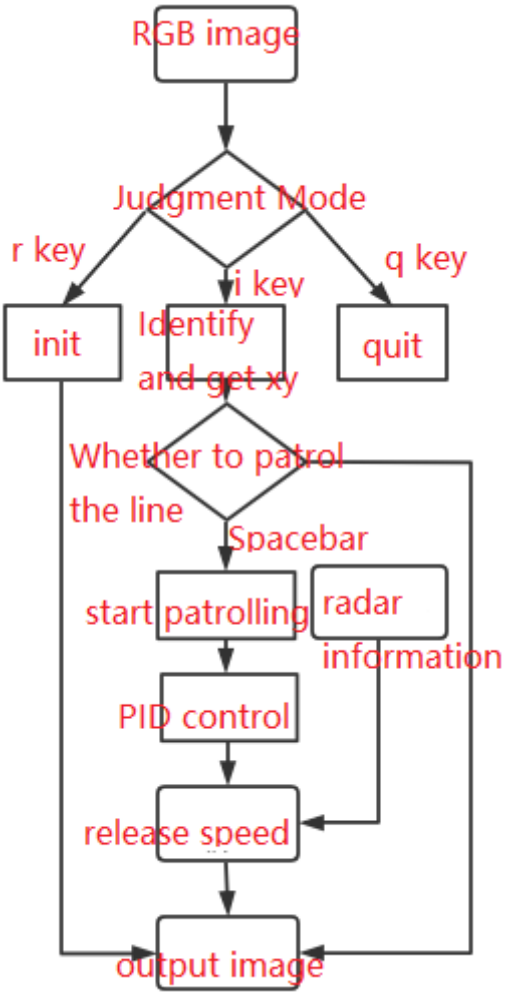
After identifying no problem, click [Spacebar] on the keyboard to execute the tracking program.

Node view

```
rqt_graph
```



[LineDetect]Node Analysis



- Subscribe to LiDAR
- Subscribe to images
- Subscription handle
- Post speed information
- Post buzzer information