2. Astra color tracking

2.1. Introduction

The Astra color tracking of the Yahboom mobile robot can recognize multiple colors at any time, store the current recognized color autonomously, control the car to follow the detected and recognized color, and keep a certain distance from the object.

The color tracking of the Yahboom mobile robot can also realize the function of real-time HSV control. By adjusting the high and low thresholds of HSV, interfering colors are filtered out, so that the blocks can be recognized very well in complex environments. If the effect of color picking is not ideal, it is necessary to move the car to different environments for calibration to achieve the recognition of the colors we need in complex environments.

2.1.1. Introduction to HSV

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

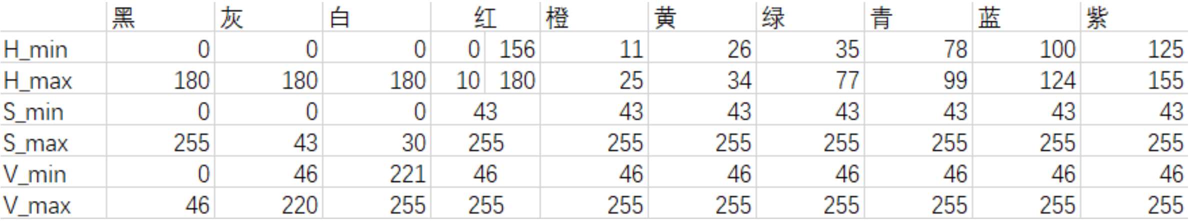
The color parameters in this model are: hue (H), saturation (S), and brightness (V).

H: 0 — 180

S: 0 — 255

V: 0 — 255

Here, part of the red is classified as the purple range:



	黑	灰	白	红		橙	黄	绿	青	蓝	紫
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

2.1.2, HSV hexagonal pyramid

- **Hue H**

Indicates color information, that is, the position of the spectral color. This parameter is expressed as an angle, ranging from 0° to 360°, starting from red and counting counterclockwise, red is 0°, green is 120°, and blue is 240°. Their complementary colors are: yellow is 60°, cyan is 180°, and purple is 300°.

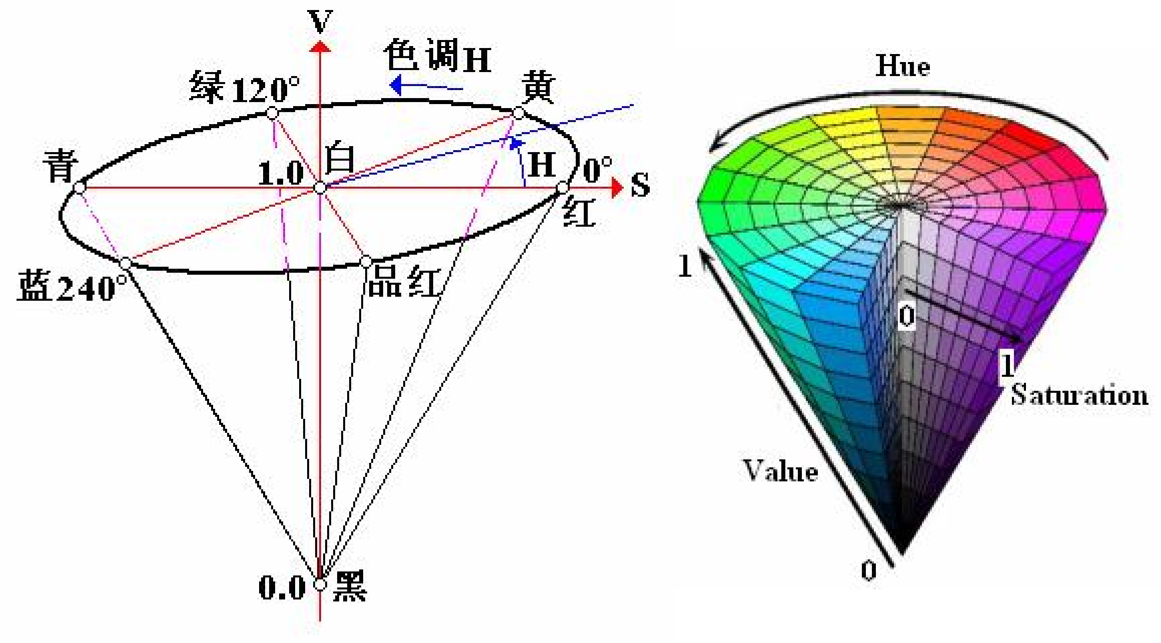
- **Saturation S**

Saturation S is expressed as the ratio between the purity of the selected color and the maximum purity of the color. When S=0, there is only grayscale. They are 120 degrees apart. The complementary colors differ by 180 degrees. A color can be seen as the result of mixing a certain spectral color with white. The greater the proportion of the spectral color, the closer the color is to the spectral color, and the higher the saturation of the color. The higher the saturation, the deeper and brighter the color. The white light component of the spectral color is 0, and the saturation reaches the highest. The value range is usually 0% to 100%. The larger the value, the more saturated the color.

- **Brightness V**

Brightness indicates the brightness of the color. For the light source color, the brightness value is related to the brightness of the light source; for the object color, this value is related to the transmittance or reflectance of the object. The value range is usually 0% (black) to 100% (white). One thing to note is that there is no direct connection between it and the light intensity.

The three-dimensional representation of the HSV model evolved from the RGB cube. Imagine observing from the white vertex along the diagonal of the RGB cube to the black vertex, you can see the hexagonal shape of the cube. The hexagonal boundary represents the color, the horizontal axis represents the purity, and the brightness is measured along the vertical axis.



2.2, Operation steps

2.2.1, Start

Note: The [SWB] mid-range of the aircraft model remote control has the [Emergency Stop] function for this gameplay. Please put the aircraft model remote control in a convenient place for control. Pay attention to safety when playing! ! !

When the image is displayed, press the [q] key to exit.

- To start control, you need to first turn the SWB button to the upper gear position (control command mode) to release the remote control

Turn off the self-starting chassis service

```
sudo supervisorctl stop ChassisServer
```

Start

```
sudo supervisorctl start LaserServer #start/stop Switch radar service (indoor version)
roslaunch yahboom_navrobo_astra colorTracker.launch VideoSwitch:=true
```

- The car has a backward collision avoidance function, and it is necessary to ensure that the radar starts normally. If you run `rostopic echo /scan`

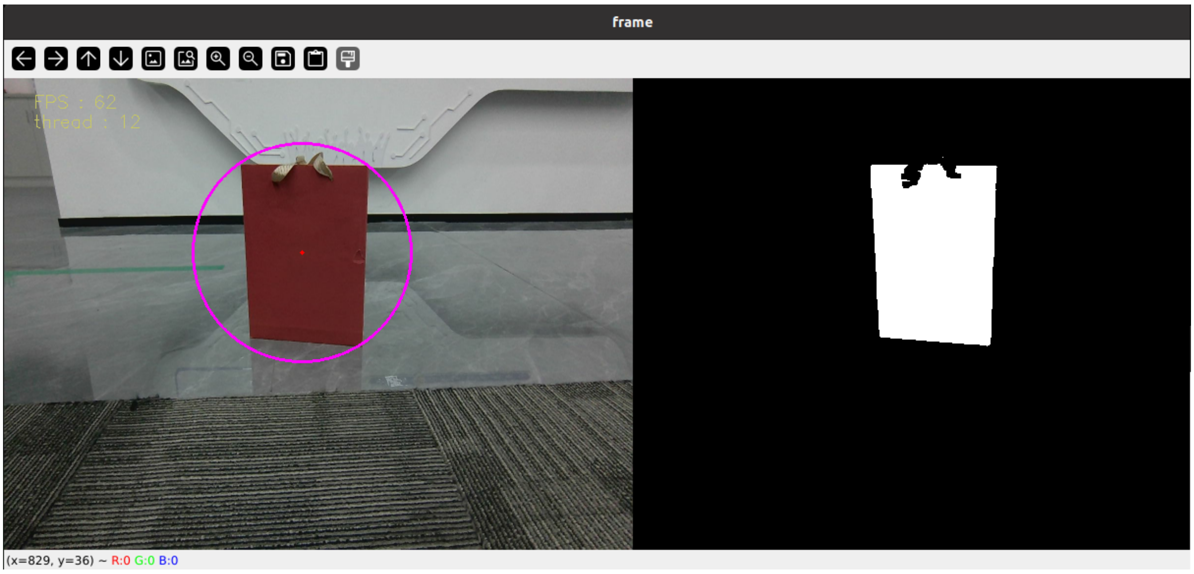
and the print is empty and no data can be obtained, the startup is abnormal. Please restart the radar service command.

- VideoSwitch parameter: whether to use the camera function package to start.

Set parameters according to needs, or directly modify the launch file, so that no additional parameters are required when starting.

2.2.2, Identification

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



Keyboard key control:

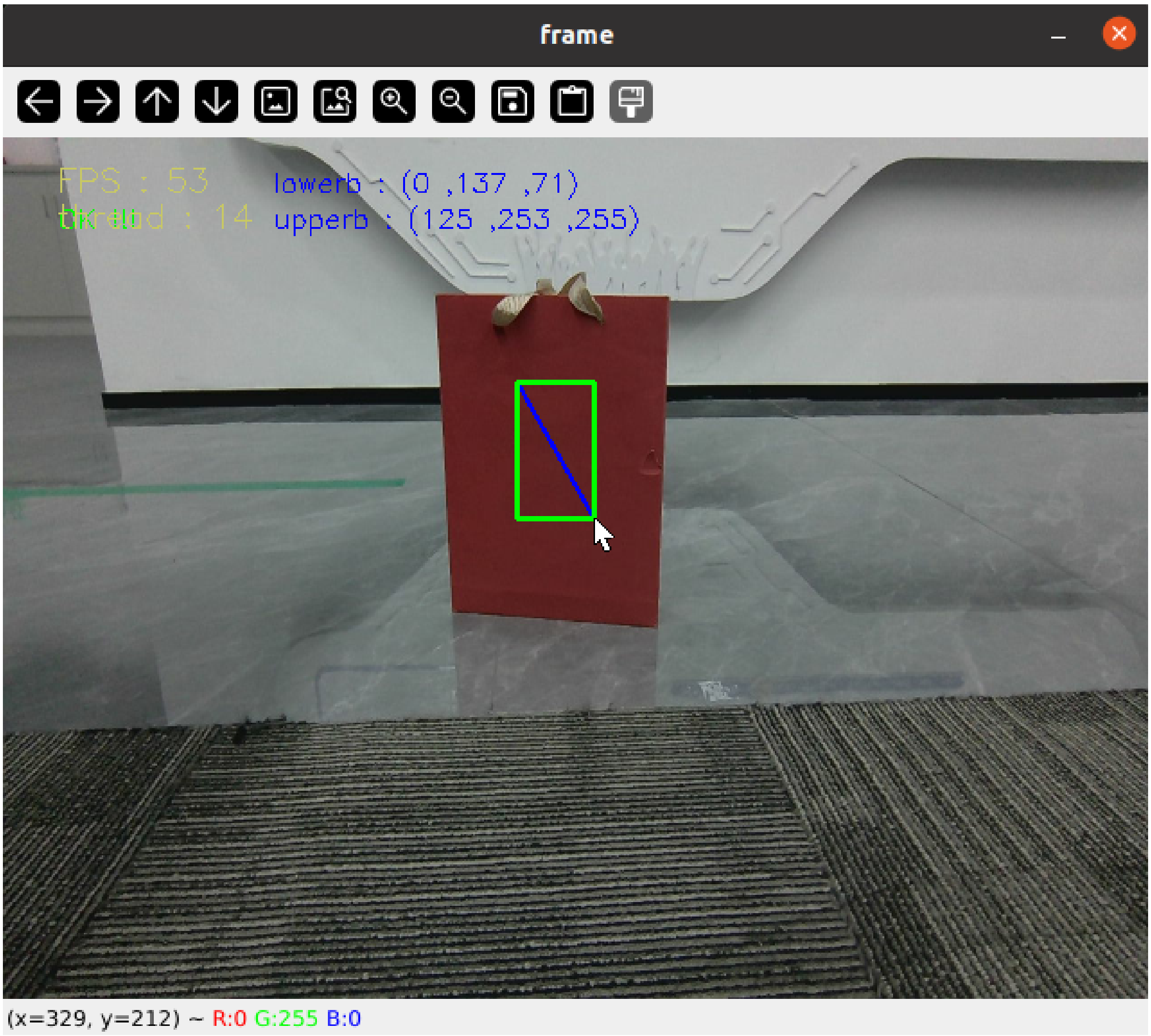
[r]: Color selection mode, you can use the mouse to select the area of the color to be identified (cannot exceed the area range).

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

[q]: Exit the program.

[Spacebar]: After there is no problem with the identification, click the keyboard [Spacebar] to execute the color following program.

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

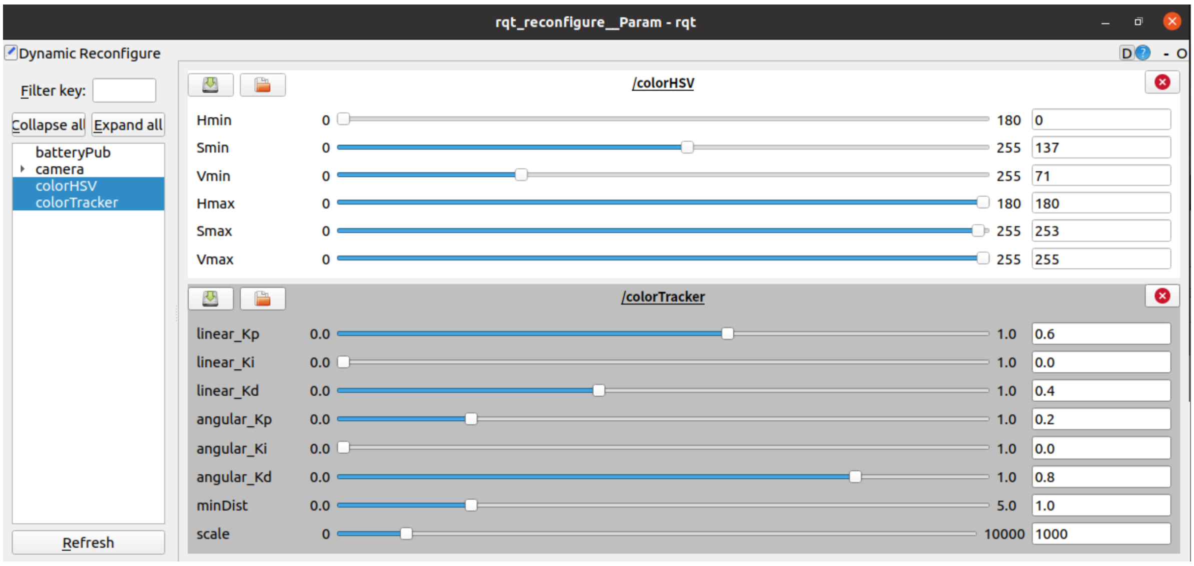


(x=329, y=212) ~ R:0 G:255 B:0

2.2.3, Color Calibration

Dynamic Parameter Adjustment

```
rosrun rqt_reconfigure rqt_reconfigure
```



/colorHSV

Hmin	0	180	0
Smin	0	255	137
Vmin	0	255	71
Hmax	0	180	180
Smax	0	255	253
Vmax	0	255	255

/colorTracker

linear_Kp	0.0	1.0	0.6
linear_Ki	0.0	1.0	0.0
linear_Kd	0.0	1.0	0.4
angular_Kp	0.0	1.0	0.2
angular_Ki	0.0	1.0	0.0
angular_Kd	0.0	1.0	0.8
minDist	0.0	5.0	1.0
scale	0	10000	1000

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

Parameter analysis:

[linear_Kp], [linear_Ki], [linear_Kd]: Linear speed PID control during the car following process.

[angular_Kp], [angular_Ki], [angular_Kd]: Angular speed PID control during the car following process.

[minDist]: Following distance, always maintain this distance.

[scale]: PID scaling.

- Parameter modification

When the parameters are adjusted to the optimal state, modify the corresponding parameters to the file, and no adjustment is required when using it again.

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

```
class color_Tracker:
def __init__(self):
rospy.on_shutdown(self.cleanup)
self.bridge = CvBridge()
self.minDist = 1500
... ...
self.linear_PID = (0.8, 0.0, 0.4)
self.angular_PID = (0.2, 0.0, 0.8)
self.scale = 1000
self.PID_init()
```

[rqt_reconfigure] Debug tool initial value modification

```
gen.add("linear_Kp", double_t, 0, "Kp in PID", 0.6, 0, 1.0) gen.add("linear_Ki",
double_t, 0, "Ki in PID", 0.0, 0, 1.0) gen.add("linear_Kd", double_t, 0, "Kd in
PID", 0.4, 0, 1.0) gen.add("angular_Kp", double_t, 0, "Kp in PID", 0.2, 0, 1 .0)
gen.add("angular_Ki", double_t, 0, "Ki in PID", 0.0, 0, 1.0)
gen.add("angular_Kd", double_t, 0, "Kd in PID", 0.8, 0, 1.0) gen.add("minDist",
double_t, 0, "minDist", 1.0, 0, gen 5.0) .add("scale", int_t, 0, "scale", 1000,
0, 10000)
exit(gen.generate(PACKAGE, "colorTracker", "ColorTrackerPID"))
```

Enter the [cfg] folder of the [yahboom_navrobo_astra] function package and modify the initial values of the corresponding parameters in the [ColorTrackerPID.cfg] file.

```
gen.add("linear_Kp", double_t, 0, "Kp in PID", 0.6, 0, 1.0)
```

Analyze the above example

Parameter	Analysis	Corresponding parameter
name	Name of the parameter	"linear_Kp"
type	Parameter data type	double_t
level	A bit mask passed to the callback	0
description	A description parameter	"Kp in PID"
default	Initial value of the node startup	0.6
min	Minimum value of the parameter	0
max	Maximum value of the parameter	1.0

Note: After the modification is completed, the environment must be recompiled and updated to be effective.

```
cd ~/YBAMR-COBOT-EDU-00001/
catkin build yahboom_navrobo_astra
source install/setup.bash
```

2.3, Program analysis

colorTracker.launch file

```
<launch>
<arg name="VideoSwitch" default="true"/>
<!-- Depth camera -->
<include file="$(find orbbec_camera)/launch/astra_pro2.launch"></include>
<!-- Chassis driver -->
<include file="$(find scout_bringup)/launch/scout_mini_robot_bringup.launch"/>
<!-- Range from the camera [0.4m, 8m], otherwise invalid -->
<!--Range from the camera [0.4m, 8m], Otherwise invalid-->
<node pkg="yahboom_navrobo_astra" type="colorTracker.py" name="colorTracker"
required="true" output="screen"/>
<node pkg="yahboom_navrobo_astra" type="colorHSV.py" name="colorHSV"
required="true" output="screen" if="$(arg VideoSwitch)">
<param name="VideoSwitch" type="bool" value="False"/>
</node>
</launch>
```

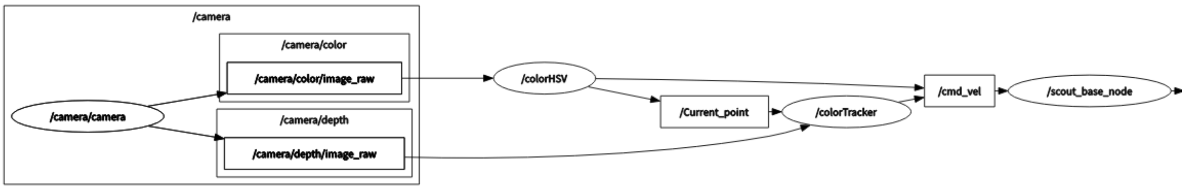
launch file analysis

If the [VideoSwitch] parameter is true, the monocular color camera is not started, and the [colorHSV] node is started. The way to obtain color images is directly implemented by the [colorHSV] node using [video*].

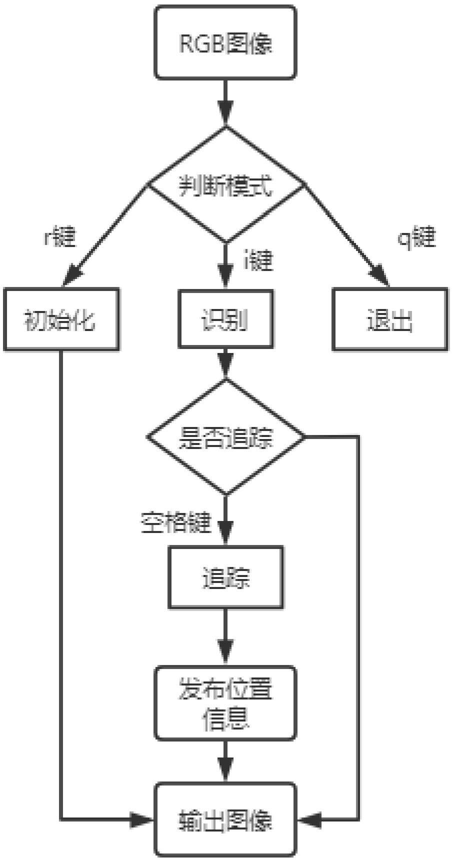
If the [VideoSwitch] parameter is false, the monocular color camera is started, and the [colorHSV】node. At this time, you need to start the 【colorHSV】node separately.

- Node View

```
rqt_graph
```

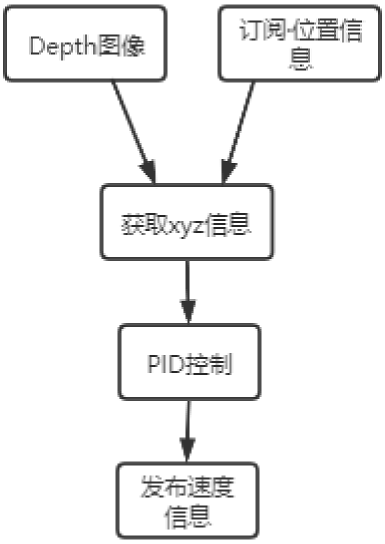


Node 【color_HSV】



- Subscribe to color image
- Publish the position of the identified object in the image
- Publish control instructions (when there is no recognized color in the picture, only the stop command is published)

Node 【color_Tracker】



- Subscribe to depth image

- Subscribe to handle control information
- Subscribe to the position information of the identified object in the image
- Publish the car follow control instruction