Voice Control Color Tracking

1. Functional Description

By interacting with the voice array module on NAVROBO, you can realize the function of opening or closing the red/blue/green/green line of NAVROBO patrol by voice.

2. Start

Note: The [SWB] mid-range of the aircraft model remote control has the [emergency stop] function of this gameplay

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

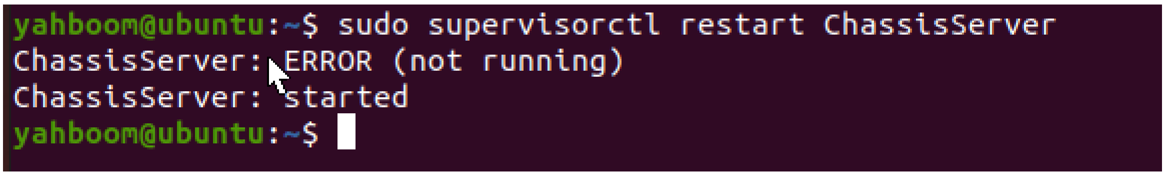
2.1. Code Path

```
/home/yahboom/wukong-robot/voice_color_identify.py
```

2.2. Start

First, close all other terminals. Open a terminal and input the following command to restart the self-starting chassis service

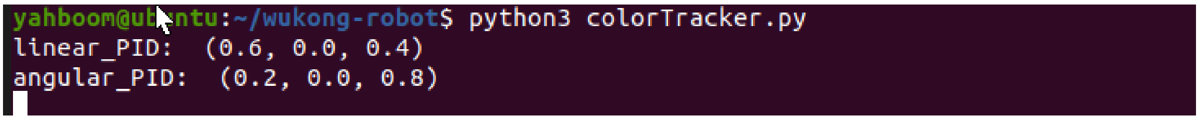
```
sudo supervisorctl restart ChassisServer
```



```
yahboom@ubuntu:~$ sudo supervisorctl restart ChassisServer
ChassisServer: ERROR (not running)
ChassisServer: started
yahboom@ubuntu:~$ 
```

To start the voice control color tracking function, enter the following command,

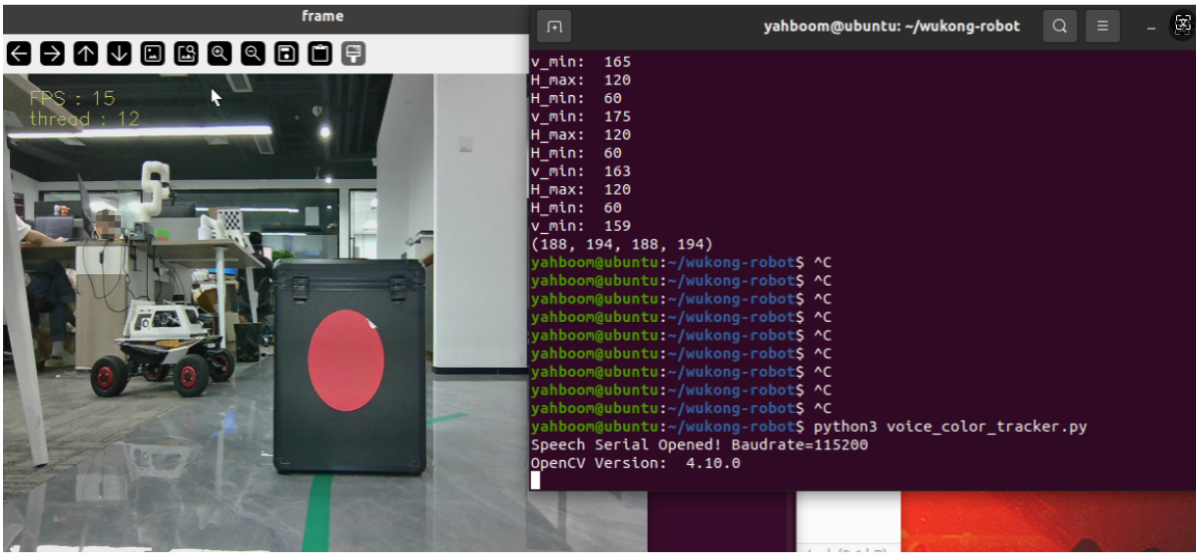
```
cd wukong-robot/
python3 colorTracker.py
```



```
yahboom@ubuntu:~/wukong-robot$ python3 colorTracker.py
linear_PID:  (0.6, 0.0, 0.4)
angular_PID:  (0.2, 0.0, 0.8)
```

Open another terminal and input,

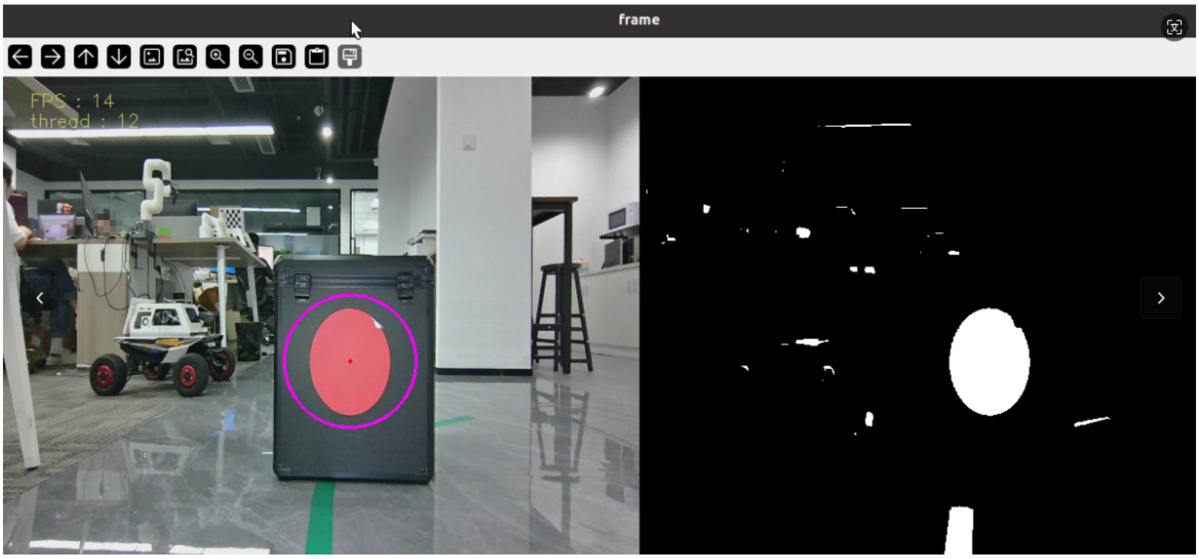
```
cd wukong-robot/
python3 voice_color_tracker.py
```



```
v_min:  165
H_max:  120
H_min:  60
v_min:  175
H_max:  120
H_min:  60
v_min:  163
H_max:  120
H_min:  60
v_min:  159
(188, 194, 188, 194)
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ ^C
yahboom@ubuntu:~/wukong-robot$ python3 voice_color_tracker.py
Speech Serial Opened! Baudrate=115200
OpenCV Version:  4.10.0
```

After the program starts, call NAVROBO "Hello Xiaoya" to wake up the module. When it broadcasts "Yes", it means the module is awakened. Taking tracking green as an example, you can then say "Start tracking green" to it, and NAVROBO will broadcast "OK, tracking green has started". Then, we turn the remote control SWB button to **upper gear**, release the control of NAVROBO, press **space bar**, and NAVROBO will start tracking green. If you don't have a remote control for the model aircraft, you can also enter the following command through the terminal,

```
rostopic pub /JoyState std_msgs/Bool True
```



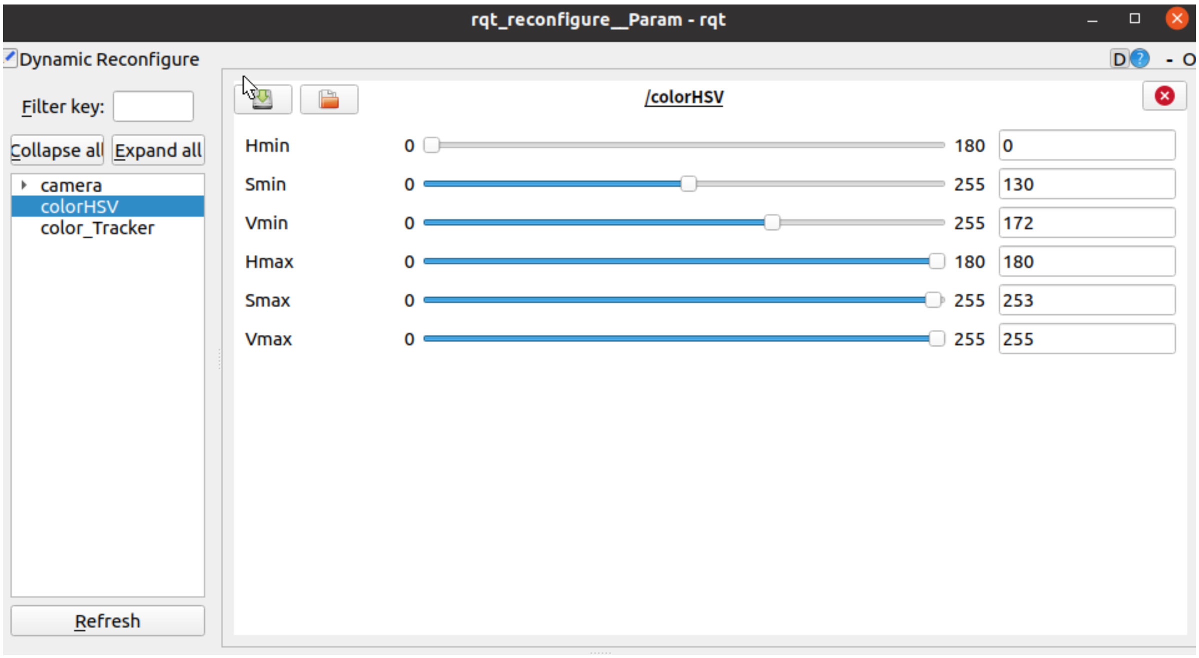
If you want to cancel the color tracking function, you can first turn the remote control SWB button to **mid-range**, then say "Xiao Ya" to NAVROBO, wake up the module, and then say "stop tracking", NAVROBO stops, and the voice will broadcast "OK, tracking has stopped".

2.3, Color Calibration

The camera is very sensitive to light, so sometimes the color recognition will be inaccurate. At this time, you need to recalibrate the red, green, yellow and blue colors. Terminal input,

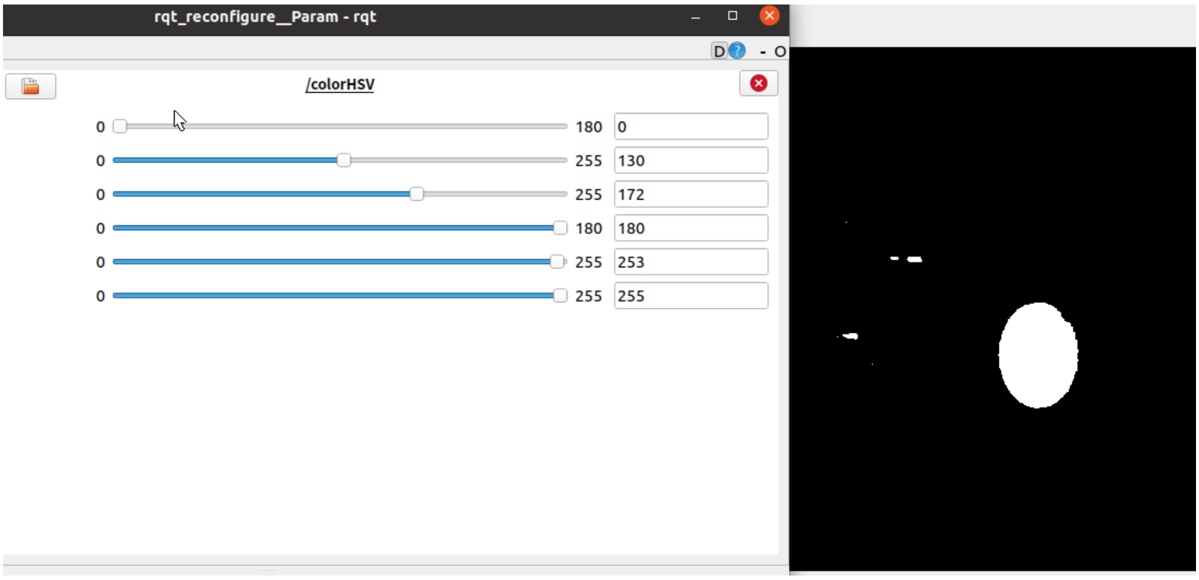
```
rosrun rqt_reconfigure rqt_reconfigure
```

Find the colorHSV column, drag the slider to modify the HSV value,



```
Hmin   0  180  0
Smin   0  255  130
Vmin   0  255  172
Hmax   0  180  180
Smax   0  255  253
Vmax   0  255  255
```

After adjustment,



```
0  180  0
0  255  130
0  255  172
0  180  180
0  255  253
0  255  255
```

Open the voice_Ctrl_color_tracker.py program and find the following part,

```
if self.command_result == 73 :
            self.model = "color_follow_line"
            print("tracker red")
            self.hsv_range =  [(0, 84, 131), (190, 253, 255)]           #[(0, 84, 131), (180, 253, 255)]
            self.dyn_update = True
            self.spe.void_write(self.command_result)
            time.sleep(0.05)

        elif self.command_result == 74 :
            self.model = "color_follow_line"
            print("tracker green")
            self.hsv_range =  [(44, 138, 91), (84, 255, 255)]
            self.dyn_update = True
            self.spe.void_write(self.command_result)
        elif self.command_result == 75 :
            self.model = "color_follow_line"
            print("tracker bule")
            self.hsv_range =  [(55, 162, 91), (125, 253, 255)]
            self.dyn_update = True
            self.spe.void_write(self.command_result)
        elif self.command_result == 72 :
            self.model = "color_follow_line"
            print("tracker yellow")
            self.hsv_range =  [(18, 55, 187), (81, 253, 255)]
            self.dyn_update = True
            self.spe.void_write(self.command_result)
        elif self.command_result == 76 :
            self.model = "Stop"
            self.spe.void_write(self.command_result)
            self.pub_cmdVel.publish(Twist())
```

Modify the HSV value after calibration just recorded to the corresponding color position, save, and use the calibrated value the next time you start.

3. Core code analysis voice_Ctrl_color_tracker.py

```
def process(self, rgb_img, action):
        rgb_img = cv.resize(rgb_img, (640, 480))
        binary = []
        # cv.setMouseCallback(self.windows_name, self.onMouse, 0)
        time.sleep(0.05)
        self.command_result = self.spe.speech_read()

        if action == 32:
            self.Track_state = 'tracking'
            rospy.loginfo("tracking")
        if action == ord('q') or action == ord('Q'): self.cancel()
        #command_result = self.spe.speech_read()
        if self.command_result == 73 :
            self.model = "color_follow_line"
            print("tracker red")
            self.hsv_range =  [(0, 84, 131), (190, 253, 255)]            #[(0, 84, 131), (180, 253, 255)]
            self.dyn_update = True
            self.spe.void_write(self.command_result)
            time.sleep(0.05)

        elif self.command_result == 74 :
            self.model = "color_follow_line"
            print("tracker green")
            self.hsv_range =  [(44, 138, 91), (84, 255, 255)]
            self.dyn_update = True
            self.spe.void_write(self.command_result)
        elif self.command_result == 75 :
            self.model = "color_follow_line"
            print("tracker bule")
            self.hsv_range =  [(55, 162, 91), (125, 253, 255)]
            self.dyn_update = True
            self.spe.void_write(self.command_result)
        elif self.command_result == 72 :
            self.model = "color_follow_line"
            print("tracker yellow")
            self.hsv_range =  [(18, 55, 187), (81, 253, 255)]
            self.dyn_update = True
            self.spe.void_write(self.command_result)
        elif self.command_result == 76 :
            self.model = "Stop"
            self.spe.void_write(self.command_result)
            self.pub_cmdVel.publish(Twist())
        self.command_result = 999
        if self.dyn_update == True :
            params = {'Hmin': self.hsv_range[0][0], 'Hmax': self.hsv_range[1][0],
                          'Smin': self.hsv_range[0][1], 'Smax': self.hsv_range[1]
[1],
                          'Vmin': self.hsv_range[0][2], 'Vmax': self.hsv_range[1]
[2]}
            self.dyn_client.update_configuration(params)
            self.dyn_update = False
        if  self.model == "color_follow_line":
            #self.ros_ctrl.Joy_active == False
            #self.model == "General"
            rgb_img, binary, self.circle = self.color.object_follow(rgb_img,
self.hsv_range)
            if self.Track_state == 'tracking':
                if self.circle[2] != 0: threading.Thread(
                target=self.execute, args=(self.circle[0], self.circle[1],
self.circle[2])).start()
                if self.point_pose[0] != 0 and self.point_pose[1] != 0:
threading.Thread(
                target=self.execute, args=(self.point_pose[0],
self.point_pose[1], self.point_pose[2])).start()
            #threading.Thread(target=self.execute, args=(self.circle[0],
self.circle[2])).start()
        return rgb_img, binary
```

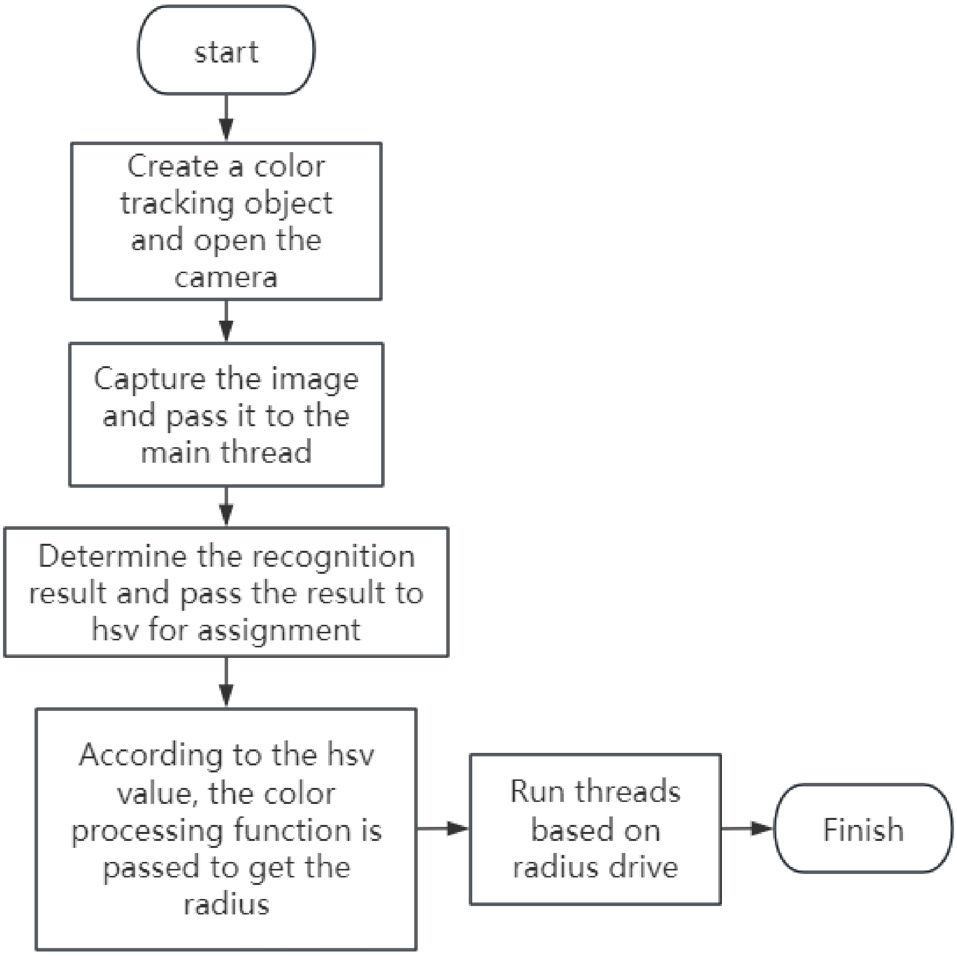
完整代码可以参考：

```
/home/yahboom/wukong-robot/colorTracker.py
/home/yahboom/wukong-robot/voice_color_tracker.py
```

3.1, Functional module communication table

Functional words	Voice recognition module results	Voice broadcast content
yellow following	72	OK, I found the yellow
red following	73	OK, I found the red
green following	74	OK, I found the green
follow this color	75	OK, I found this color

3.2, Program Flowchart


start
Create a color tracking object and open the camera
Capture the image and pass it to the main thread
Determine the recognition result and pass the result to hsv for assignment
According to the hsv value, the color processing function is passed to get the radius
Run threads based on radius drive
Finish