6. Voice control color tracking

6.1. Function description

By interacting with the voice recognition module on ROSMASTER, you can enable or disable the function of ROSMASTER tracking red/blue/green/yellow color. The R2 button on the handle can cancel/enable this function at any time.

6.2. start

6.2.1. function package path

```
~/yahboomcar/src/yahboomcar_voice_ctrl/
```

6.2.2. Start

```
roslaunch yahboomcar_voice_ctrl voice_ctrl_colorTracker.launch
python
~/yahboomcar_ws/src/yahboomcar_voice_ctrl/scripts/voice_Ctrl_color_tracker.py
```

After the program starts, call "Hi Yahboom" to ROSMASTER to wake up the module, when it broadcasts "Hi , I'm here.", it means to wake up the module. Take tracking red as an example, then you can say "red following" to it, and ROSMASTER will broadcast "OK, I found the red". Then, we release the control of ROSMASTER by pressing the R2 button of the handle, and ROSMASTER starts to track red. If there is no remote control, you can also enter the following commands through the terminal,

```
rostopic pub /JoyState std_msgs/Bool False
```

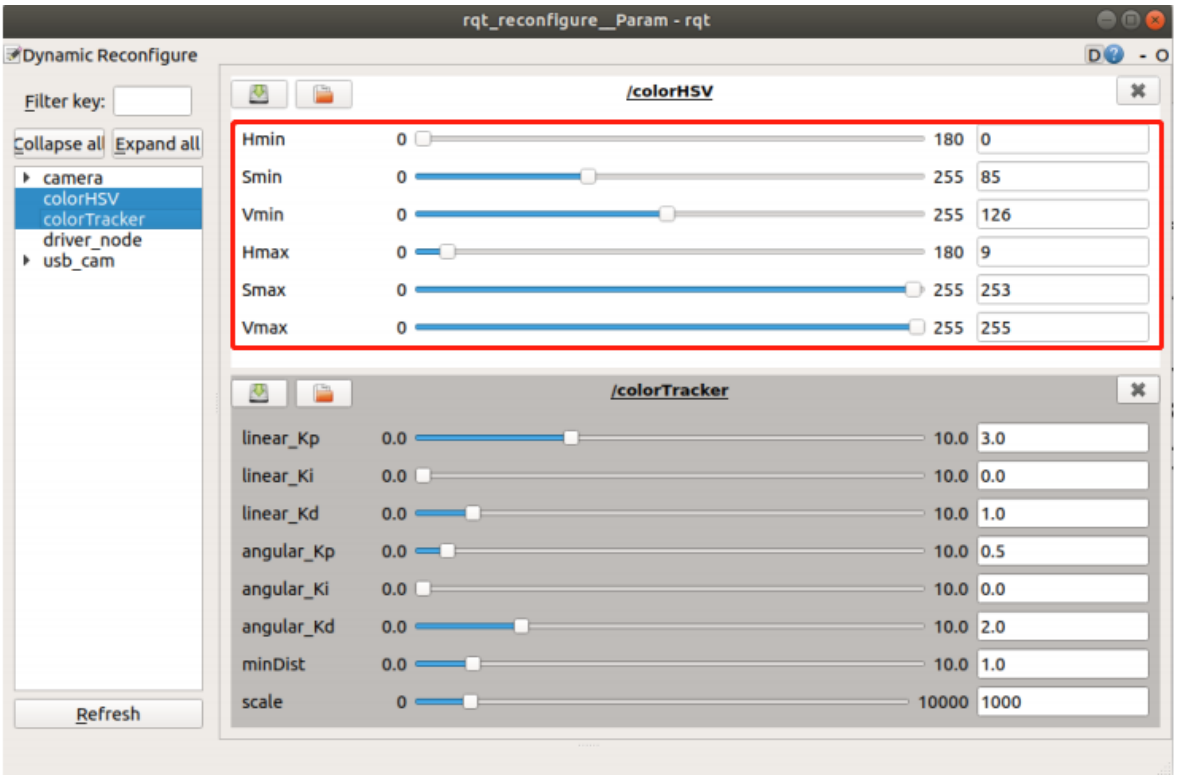
If you want to cancel the color tracking function, say "stop following" to ROSMASTER, ROSMASTER will stop, and the voice will broadcast "OK, it has been stoped".

6.2.3. Color calibration

The camera is very sensitive to light, so sometimes the color recognition will be inaccurate. At this time, it is necessary to re-calibrate the colors of red, green, yellow and blue. terminal input,

```
rosrun rqt_reconfigure rqt_reconfigure
```

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



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

```
if command_result == 73 :
    self.model = "color_follow_line"
    print("tracker red")
    self.hsv_range =  [(0, 185, 175), (180, 253, 255)]
elif command_result == 74 :
    self.model = "color_follow_line"
    print("tracker green")
    self.hsv_range =  [(54, 92, 75), (125, 255, 255)]
elif command_result == 75 :
    self.model = "color_follow_line"
    print("tracker bule")
    self.hsv_range =  [(55, 204, 177), (125, 253, 255)]
elif command_result == 72 :
    self.model = "color_follow_line"
    print("tracker yellow")
    self.hsv_range =  [(18, 128, 168), (125, 253, 255)]
```

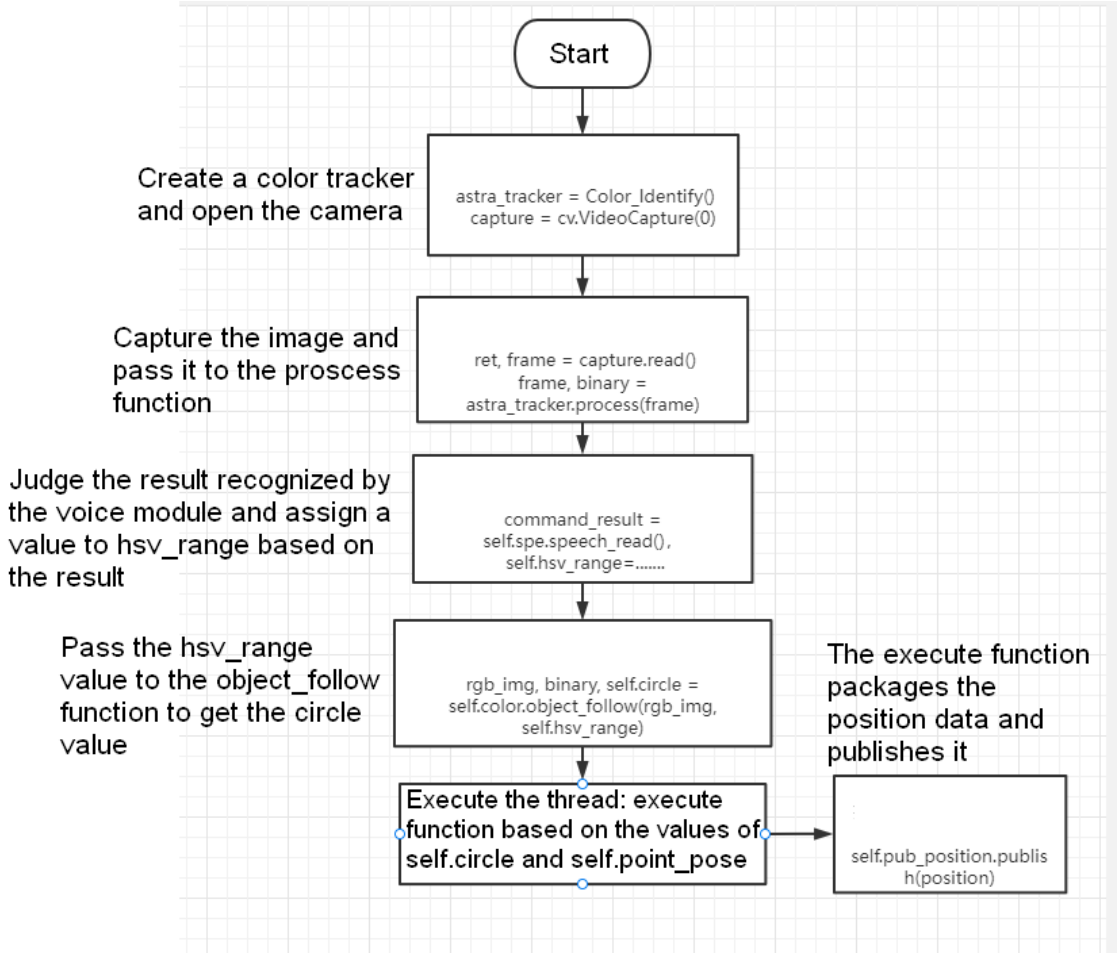
Modify the calibrated HSV value recorded just now to the position of the corresponding color, save it, and use the calibrated value the next time it is started.

6.3. Core code analysis voice_Ctrl_color_tracker.py

```
        command_result = self.spe.speech_read()
        self.spe.void_write(command_result)
        if command_result == 73 :
            self.model = "color_follow_line"
            print("tracker red")
            self.hsv_range =  [(20, 215, 111), (180, 253, 255)]
            self.dyn_update = True
```

```
        elif command_result == 74 :
            self.model = "color_follow_line"
            print("tracker green")
            self.hsv_range =  [(44, 138, 91), (84, 255, 255)]
            self.dyn_update = True
        elif command_result == 75 :
            self.model = "color_follow_line"
            print("tracker bule")
            self.hsv_range =  [(83, 217, 196), (141, 253, 255)]
            self.dyn_update = True
        elif command_result == 72 :
            self.model = "color_follow_line"
            print("tracker yellow")
            self.hsv_range =  [(18, 55, 187), (81, 253, 255)]
            self.dyn_update = True
        elif command_result == 76 :
            self.model = "Stop"
            #self.ros_ctrl.Joy_active == False
            #self.ros_ctrl.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.ros_ctrl.Joy_active == False :
                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
        def execute(self, x, y, z):
            position = Position()
            position.angleX = x
            position.angleY = y
            position.distance = z
            self.pub_position.publish(position)
```

6.3.1. Program flow chart



The complete code can refer to:

```
~/yahboomcar/src/yahboomcar_voice_ctrl/scripts/voice_Ctrl_color_tracker.py
```

6.4. Function module communication table

function word	Speech 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
stop following	76	OK, it has been stoped