Voice control color recognition

1. Functional description

By interacting with the voice array module on NAVROBO, the function of voice opening or closing the selected color recognition of the NAVROBO camera area can be realized

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. Function package path

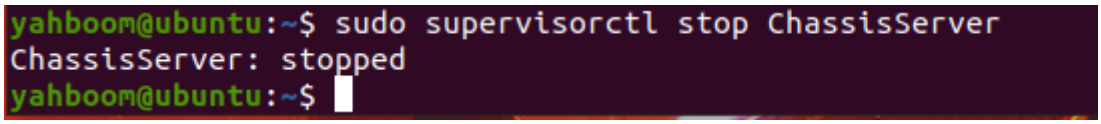
```
/home/yahboom/wukong-robot/voice_color_identify.py
```

2.2. Start

Turn off the self-start chassis service

This is because the self-start chassis occupies the camera, which needs to be released here.

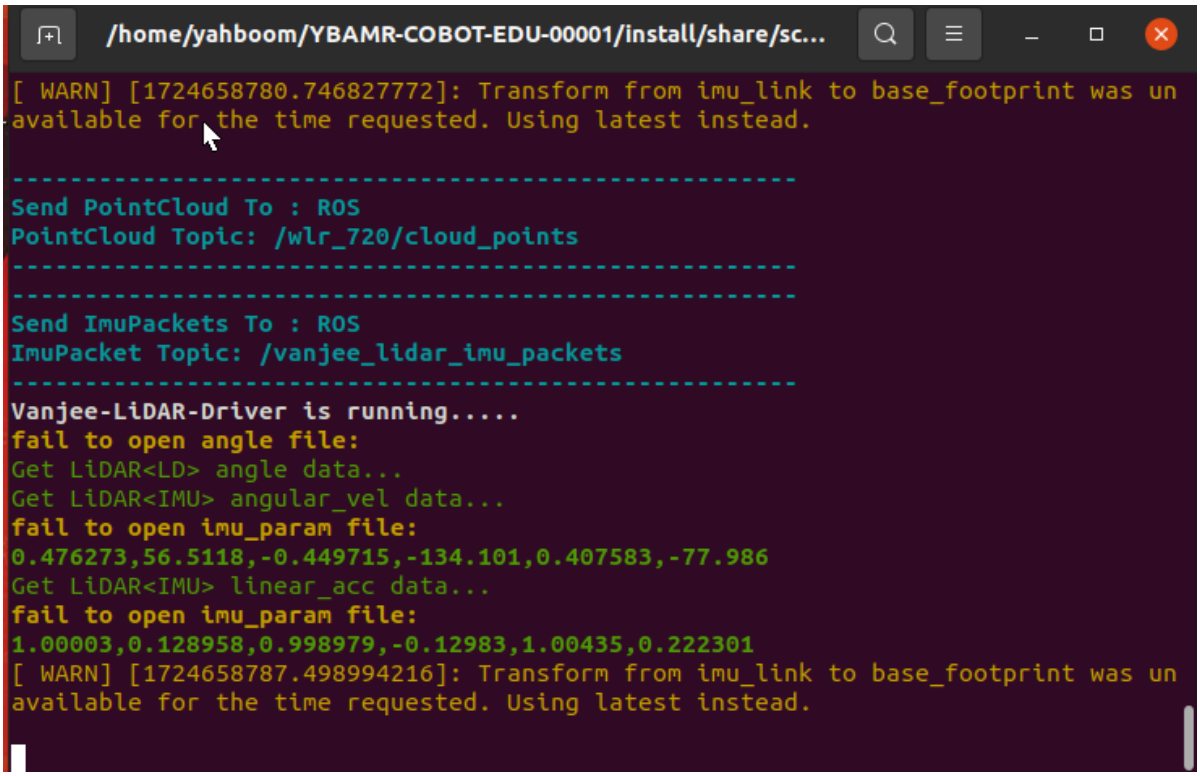
```
sudo supervisorctl stop ChassisServer
```



Manually start chassis + radar service

```
roslaunch scout_bringup navrobot_base.launch
```

The appearance of imu and baselink conversion alarms will not affect normal use, because the time does not match.



If video6 is not recognized, you can plug and unplug the USB port of the depth camera at this time,

```
ls /dev/video*
```

After plugging and unplugging, the following videos are displayed, indicating the normal status.

```
yahboom@ubuntu:~$ ls /dev/video*
/dev/video0  /dev/video2  /dev/video4  /dev/video6
/dev/video1  /dev/video3  /dev/video5  /dev/video7
yahboom@ubuntu:~$
```

Open another terminal and enter the command to start voice control color recognition.

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

```
yahboom@ubuntu:~/wukong-robot$ python3 voice_color_identify.py
Speech Serial Opened! Baudrate=115200
(260, 126, 260, 126)
(260, 126, 262, 126)
(260, 126, 266, 128)
H_max:  60
H_min:  60
v_min:  255
(260, 126, 274, 138)
(260, 126, 275, 138)
H_max:  120
H_min:  60
v_min:  180
(260, 126, 276, 141)
(260, 126, 277, 141)
H_max:  120
H_min:  60
v_min:  181
H_max:  120
```

After the program is running, place the object whose color needs to be identified in front of the camera, say "Hello, Xiaoya" to the voice module, wait until the voice module responds with "Yes", then ask it "What color is this", select the area of the object within 3 seconds, hold down the mouse and do not release it, then it will broadcast the color of the object area selected by the mouse.

Note: Since the camera is sensitive to light, the same color will appear different under different light intensities, resulting in different color recognition results.

3. Core code analysis

3.1. Import the voice recognition library and create a voice recognition object

```
self.voice_cmd_callback,queue_size=1)
...
def voice_cmd_callback(self, msg):
if msg != 0:
self.command_result = msg.data
```

3.2. Get mouse events and specify the mouse selected area

```
def onMouse(self, event, x, y, flags, param):
if event == 1:
self.select_flags = True
self.Mouse_XY = (x,y)
if event == 4:
self.select_flags = False
if self.select_flags == True:
self.cols = min(self.Mouse_XY[0], x), min(self.Mouse_XY[1], y)
self.rows = max(self.Mouse_XY[0], x), max(self.Mouse_XY[1], y)
self.Roi_init = (self.cols[0], self.cols[1], self.rows[0], self.rows[1])
print(self.Roi_init)
```

This step is mainly to get the value of self.Roi_init, which is used to get the HSV value of the area

3.3, Get the HSV value of the selected area

```
if self.Roi_init[0]!=self.Roi_init[2] and self.Roi_init[1]!=self.Roi_init[3]:
HSV = cv.cvtColor(rgb_img,cv.COLOR_BGR2HSV)
for i in range(self.Roi_init[0], self.Roi_init[2]):
for j in range(self.Roi_init[1], self.Roi_init[3]):
H.append(HSV[j, i][0])
S.append(HSV[j, i][1])
V.append(HSV[j, i][2])
H_min = min(H); H_max = max(H)
S_min = min(S); S_max = 253
V_min = min(V); V_max = 255
```

3.4. Determine the area where the HSV value is located and report the recognition result according to the interval

```
if self.command_result !=999:
print(self.command_result)
if self.command_result == 60:

if H_max > 168 and S_min > 100:
#self.spe.void_write(61)
print("red")
self.spe.void_write(61) time.sleep(0.05) elif H_min >= 22 and H_min <= 56 and
S_min > 100: print("yellow") self.spe.void_write(64) time.sleep(0.05)
#self.spe.void_write(64) elif H_min >= 60 and (S_min >110 and S_ min < 200) and
V_min > 100 : print("blue") self.spe.void_write(62) time.sleep(0.05) elif H_min
>= 56 and S_min < 160 and V_min < 120 : print("green") self.spe.void_write(63)
time.sleep(0.05) #self.spe.void_write(63)
else :
print("none")
```

For the complete code, please refer to:

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

3.5, Voice array communication table

Function words	Voice array module results	Voice broadcast content
What color is this	60	Broadcast according to the recognized color, see the table below.

Color	Host sends recognition result	Voice broadcast content
Red	61	This is red
Blue	62	This is blue
Green	63	This is green
Yellow	64	This is yellow

3.6、flow chart