

Color following

1. Experimental purpose

Drive the car to follow the color

2. Experimental path source code

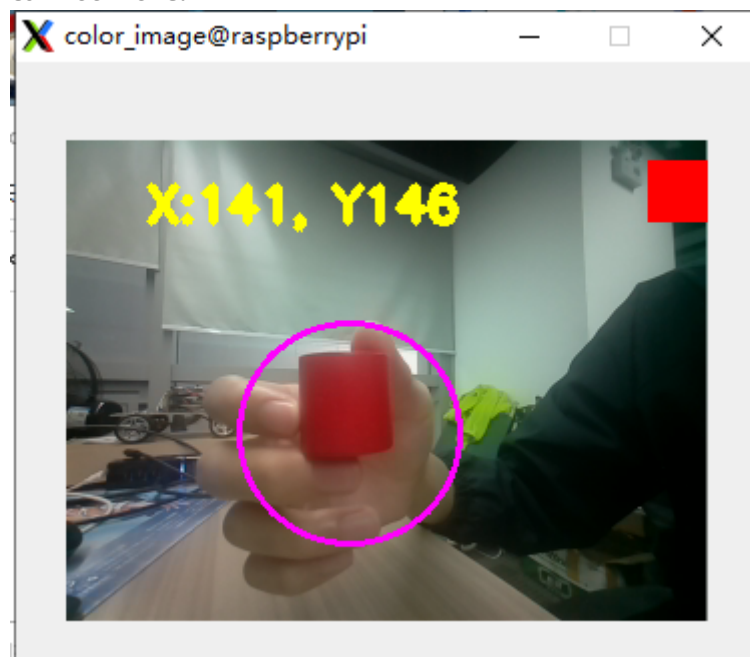
Enter the car system and end the car program
Then go to the terminal and enter

```
cd /home/pi/Rider-pi_class/6.AI Visual Interaction Course/2. Color following/  
python3 color_follow.py
```

3. Experimental phenomenon

After running the source code, the car can identify one of the four colors **red, yellow, blue and green**. You can press the button in the upper right corner of the car screen to switch the color to follow.

The car cannot be in a stopped state (that is, not in a standing balance state), otherwise it cannot move.



4. Analysis of main source code parameters

```
#-----COMMON INIT-----  
  
font = cv2.FONT_HERSHEY_SIMPLEX  
cap=cv2.VideoCapture(0)  
cap.set(3,320)  
cap.set(4,240)  
if(not cap.isOpened()):  
    print("[camera.py:cam]:can't open this camera")
```

```

t_start = time.time()
fps = 0
color_x = 0
color_y = 0
color_radius = 0
while 1:
    ret, frame = cap.read()
    frame_ = cv2.GaussianBlur(frame, (5, 5), 0)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv, color_lower, color_upper)
    mask = cv2.erode(mask, None, iterations=2)
    mask = cv2.dilate(mask, None, iterations=2)
    mask = cv2.GaussianBlur(mask, (3, 3), 0)
    cnts =
cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[-2]
    if g_mode == 1:
        if len(cnts) > 0:
            cnt = max(cnts, key = cv2.contourArea)
            (color_x, color_y), color_radius = cv2.minEnclosingCircle(cnt)
            if color_radius > 10:
                cv2.circle(frame, (int(color_x), int(color_y)), int(color_radius),
(255, 0, 255), 2)
                value_x = color_x - 160
                value_y = color_y - 120
                rider_x=value_x

                value_radius = color_radius - 40 #面积area
                if value_x > 55:
                    value_x = 55
                elif value_x < -55:
                    value_x = -55
                if value_y > 75:
                    value_y = 75
                elif value_y < -75:
                    value_y = -75

                #如果是小车If it is a small car
                if car_type=='R':
                    # print(color_y, 75+int((190-color_y)/160*40))
                    # g_car.rider_height(75+int((190-color_y)/160*40))
                    if value_radius == 9999:
                        g_car.rider_move_x(0)
                    else:
                        if value_radius > 15:
                            #print("back"+str(value_radius))
                            g_car.rider_move_x(-0.3) #后退 #0.3太快了Back #0.3 Too
fast
                        elif value_radius < -15:
                            #print("go"+str(value_radius))
                            g_car.rider_move_x(0.1) #前进 #0.3太快了Go forward
#0.3 too fast
                        else:
                            g_car.rider_move_x(0)

```

```

        if rider_x==9999:
            g_car.rider_turn(0)
        else:
            if rider_x > 35:
                g_car.rider_turn(-30)
            elif rider_x < -35:
                g_car.rider_turn(30)
            else:
                g_car.rider_turn(0)
    else:
        color_x = 0
        color_y = 0
        rider_x=9999
        #print("stop")
        g_car.rider_move_x(0)
        g_car.rider_turn(0)
        cv2.putText(frame, "x:%d, y%d" % (int(color_x), int(color_y)), (40,40),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 3)
        t_start = time.time()
        fps = 0
    else:
        fps = fps + 1
        mfps = fps / (time.time() - t_start)
        cv2.putText(frame, "FPS " + str(int(mfps)), (40,40),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 3)

b,g,r = cv2.split(frame)
img = cv2.merge((r,g,b))
if mode==1:
    cv2.rectangle(img, (290, 10), (320, 40), red, -1)
elif mode==2:
    cv2.rectangle(img, (290, 10), (320, 40), green, -1)
elif mode==3:
    cv2.rectangle(img, (290, 10), (320, 40), blue, -1)
elif mode==4:
    cv2.rectangle(img, (290, 10), (320, 40), yellow, -1)
imgok = Image.fromarray(img)
display.ShowImage(imgok)

r,g,b = cv2.split(img)
frame1 = cv2.merge((b,g,r))
cv2.imshow("color_image", frame1) #同时显示在屏幕上Displayed on screen at the
same time

# if (cv2.waitKey(1)) == ord('q'):
#     break
if (cv2.waitKey(1)) == ord('f'): #Switch Color
    change_color()
if button.press_b():
    break
if button.press_d():
    change_color()

```

```
cap.release()
```

From the source code analysis, we can see that the car will follow the color according to the camera. If the color is forward, the car will move forward; if the color is backward, the car will move backward; if the color is left or right, the car will also follow left or right. And the car screen and computer screen will display the corresponding recognition results.