

Chapter16: Raspberry pi face tracking

In the previous course, we have learned how realize face recognition, alarming and the color tracking with servo. In this lesson, we will combine several techniques to complete a face tracking program with servo control.

uses the PCA9685 and Raspberry Pi to control the servo and the color recognition. In this lesson, we will try to combine these two technologies.

The camera PTZ is controlled by two servos. The PID technology makes the movement of the servos stable, and the masks in the color tracking, morphological processing and other commands are replaced to the recognition of the [face.xml](#) file.

The source code of the program is located at:

[/home/pi/Adafruit_Python_PCA9685/servo_face_nosocket.py](#)

```

1 #!/usr/bin/env python2
2 # -*- coding: utf-8 -*-
3 """
4 * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
5 * @file      sevo_face_nosocket
6 * @version   V1.0
7 * @details
8 * @par History
9 * @author: longFuSun
10 """
11
12 from __future__ import division
13 import cv2
14 import Adafruit_PCA9685
15 import time
16 import numpy as np
17 import threading
18 #初始化PCA9685和舵机
19 pwm = Adafruit_PCA9685.PCA9685()
20 pwm.set_pwm_freq(60)
21 pwm.set_pwm(1,0,500)
22 pwm.set_pwm(2,0,500)
23 time.sleep(1)
24 #Initialize the camera and set the threshold
25 #If you think it is stuck, please adjust the "1" and "2" two codes
26 cap = cv2.VideoCapture(0)
27 #“1”, Camera resolution, center point is (320, 240)
28
29 cap.set(cv2.cv.CV_CAP_PROP_FOURCC, cv2.cv.CV_FOURCC('M', 'J', 'P', 'G'))
30 cap.set(3, 320)
31 cap.set(4, 240)
32 #Import Classifier
33 face_cascade = cv2.CascadeClassifier( '123.xml' )
34 x=0;
35 thisError_x=0
36 lastError_x=0
37 thisError_y=0
38 lastError_y=0
39
40 Y_P = 425
41 X_P = 425
42 flag=0
43 y=0
44 w=0
45 h=0
46 facebool = False
47
48 def xx():
49     while True:
50         CON=0
51         if CON==0:
52             pwm.set_pwm(1,0,650-X_P+200)
53             #pwm.set_pwm(2,0,650-Y_P+200)
54             CON+=1
55         else:
56             pwm.set_pwm(1,0,650-X_P)
57             #pwm.set_pwm(2,0,650-Y_P)
58
59
60 tid=threading.Thread(target=xx)
61 tid.setDaemon(True)
62 tid.start()
63
64 while True:
65     ret,frame = cap.read()
66
67     #frame=cv2.GaussianBlur(frame,(5,5),0)
68     gray= cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
69     #对灰度图进行.detectMultiScale()
70     faces=face_cascade.detectMultiScale(gray)
71     max_face=0
72     value_x=0
73
74
75
76     if len(faces)>0:
77         #print('face found!')

```

```

78     #temp = (x,y,w,h)
79     (x,y,w,h) = faces[0]
80     cv2.rectangle(frame, (x,y), (x+h,y+w), (0,255,0),2)
81     result=(x,y,w,h)
82     x=result[0]+w/2
83     y=result[1]+h/2
84     facebool = True
85     '''
86
87     for(x,y,w,h) in faces:
88         #Find the center of the rectangle
89         cv2.rectangle(frame, (x,y), (x+h,y+w), (0,255,0),2)
90         result=(x,y,w,h)
91         x=result[0]+w/2
92         y=result[1]+h/2
93         '''
94
95     # "2", error value
96
97
98     #while facebool:
99         thisError_x=x-160
100        thisError_y=y-120
101        #if thisError_x > -20 and thisError_x < 20 and thisError_y > -20 and th
102        #    facebool = False
103        #he user can adjust the values of P and values of ,and detect the influ
104        pwm_x = thisError_x*5+1*(thisError_x-lastError_x)
105        pwm_y = thisError_y*5+1*(thisError_y-lastError_y)
106        lastError_x = thisError_x
107        lastError_y = thisError_y
108        XP=pwm_x/100
109        YP=pwm_y/100
110        X_P=X_P+int(XP)
111        Y_P=Y_P+int(YP)
112        if X_P>670:
113            X_P=650
114
115        if X_P<0:
116            X_P=0
117        if Y_P>650:
118            Y_P=650
119        if X_P<0:
120            X_P=0
121
122
123        #pwm.set_pwm(1,0,650-X_P)
124        #pwm.set_pwm(2,0,650-Y_P)
125
126        cv2.imshow("capture", frame)
127        if cv2.waitKey(1)==119:
128            break
129
130    cap.release()
131    cv2.destroyAllWindows()
132

```