

实验一 基于OpenCV的颜色特征识别

SA18225293 彭辰铭

配置环境与运行

本次实验原则上是可以在win10, linuxOs上进行的。

- 1、我先在Ubuntu下运行了。采取的是python2.7+opencv3.4.4。
- 2、在win10下由于采用的是python3.6+opencv3.4.4。所以需要将提供的python2.7 版本代码迁移到python3中来。

我采用的是2to3工具。

主要不兼容的地方在于python3中不再支持函数中写入元组参数解包

使用

```
1 pip install 2to3
2 //cd 到代码所在目录
3 2to3 test.py video.py common.py
4 pip install opencv-python
```

之后确保安装有所需要的包，即可正常运行

若提示缺少包，则可以使用pip工具逐一进行安装。

主要代码分析

test.py代码

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  import numpy as np
5  import cv2
6  import serial
7
8  # from picamera.array import PiRGBArray
9  # from picamera import PiCamera
10 # import time
11 # import copy
12 import video
13 from matplotlib import pyplot as plt
14
15 class App(object):
16     def __init__(self, color):
17         #调用video的create_capture函数创建cam窗口
18         self.cam = video.create_capture(0)
```

```

19 #         self.cam = PiCamera()
20 #         self.cam.resolution = (320,240)
21 #         self.cam.framerate = 32
22 #         self.rCa = PiRGBArray(self.cam, size=(320,240))
23 #         time.sleep(0.1)
24 #         self.cam.capture(self.rCa, format='bgr')
25 #         self.frame = self.rCa.array
26 #         ret, self.frame = self.cam.read()
27 #命名窗口为 camshift
28         cv2.namedwindow('camshift')
29 #color为0 表示抓取红色
30         int(color)
31         if color == 0:
32             print("yes")
33             self.roi = cv2.imread( 'hong.jpg' )
34             self.flag = "Hong"
35 #
36             self.selection = None
37             self.tracking_state = 0
38             self.show_backproj = False
39 #
40         #抓取蓝色
41         else:
42             print("no")
43             self.flag = "Lan"
44             self.roi = cv2.imread('lan.jpg')
45             self.selection = None
46             self.tracking_state = 0
47             self.show_backproj = False
48 #         self.ser = serial.Serial('/dev/ttyAMA0',115200,timeout=0.5)
49
50
51     def start(self):
52         self.tracking_state = 0
53         #x, y = np.int16([220, 110]) # BUG
54         if self.flag == 'Hong':
55             self.selection = (4, 6, 407, 304)
56             self.tracking_state = 1
57             print("yes")
58         else:
59             self.selection = (40, 54, 296, 230)
60             self.tracking_state = 1
61         #         print "start"
62
63     def show_hist(self):
64         #显示直方图
65         bin_count = self.hist.shape[0]
66         bin_w = 24
67         #初始化256各 bin_conut x 24, 3 的 矩阵 3维数组, 数据类型为np.uint8
68         img = np.zeros((256, bin_count*bin_w, 3), np.uint8)
69         for i in range(bin_count):
70             h = int(self.hist[i])

```

```

71         cv2.rectangle(img, (i*bin_w+2, 255), ((i+1)*bin_w-2, 255-h),
(int(180.0*i/bin_count), 255, 255), -1)
72         img = cv2.cvtColor(img, cv2.COLOR_HSV2BGR)
73         cv2.imshow('hist', img)
74
75     def run(self):
76         roi = self.roi
77         self.start()
78         while True:
79             # for frame in self.cam.capture_continuous(self.rCa, format='bgr',
use_video_port=True):
80             #读取视频帧
81                 ret, self.frame = self.cam.read()
82             # self.frame = frame.array
83                 vis = self.frame.copy()
84             # vis = copy.deepcopy(self.frame)
85             #把图像从rgb转换为hsv颜色空间, 色调 饱和度 亮度
86                 hsv = cv2.cvtColor(self.frame, cv2.COLOR_BGR2HSV)
87             #函数去除阈值, 去除背景部分
88                 mask = cv2.inRange(hsv, np.array((0., 60., 32.)), np.array((255.,
255., 255.)))
89             # self.selection = 1
90
91             if self.selection:
92                 # x0, y0, x1, y1 = 220, 110, 358, 245
93                 x0, y0, x1, y1 = self.selection
94                 self.track_window = (x0, y0, x1-x0, y1-y0)
95                 # hsv_roi = hsv[y0:y1, x0:x1]
96                 # mask_roi = mask[y0:y1, x0:x1]
97                 hsv_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
98                 mask_roi = cv2.inRange(hsv_roi, np.array((0., 0., 0.)),
np.array((255., 255., 255.)))
99                 #一维直方图参数设置
100                 #opencv HSV的H显示范围为0-180
101                 hist = cv2.calcHist( [hsv_roi], [0], mask_roi, [32], [0, 180] )
102                 #二维直方图参数设置
103                 #hist = cv2.calcHist( [hsv_roi], [0,2],None, [180,256], [0, 180,0
, 255] )
104                 #hist = cv2.calcHist( [hsv_roi], [0, 2],None, [180,256], [0, 180,
0, 255] )
105                 #将绘制出来的直方图归一化, 设定显示范围
106                 cv2.normalize(hist, hist, 0, 255, cv2.NORM_MINMAX);
107                 self.hist = hist.reshape(-1)
108                 #二维直方图显示
109                 # plt.imshow(hist,interpolation = 'nearest')
110                 # plt.show()
111                 # plt.imshow(hist,interpolation = 'nearest')
112                 # plt.show()
113                 #显示出直方图
114                 self.show_hist()
115
116                 vis_roi = vis[y0:y1, x0:x1]

```

```

117 #bitwise_not是对二进制数据进行“非”操作，即对图像（灰度图像或彩色图像均可）每个像素值进行二进
    制“非”操作，~1=0, ~0=1
118         cv2.bitwise_not(vis_roi, vis_roi)
119         vis[mask == 0] = 0
120
121         if self.tracking_state == 1:
122             self.selection = None
123     #反向投影
124     #BackProjection中存储的数值代表了测试图像中该像素属于红色、蓝色区域的概率
125     #这里是图像函数，channel,掩模图像，像素值范围，（可选输出反向投影的比例因子） 像素是否均匀分布
126         prob = cv2.calcBackProject([hsv], [0], self.hist, [0, 180], 1)
127     #与掩码相与，去除不需要的地方
128         prob &= mask
129     #设置迭代的终止标准，最多十次迭代
130         term_crit = ( cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10,
131 1 )
132         track_box, self.track_window = cv2.CamShift(prob,
133 self.track_window, term_crit)
134     #         if track_box[0][1] <= 240:
135     #             self.ser.write(str(int(track_box[0][0])-320) + " " +
136 str(int(track_box[0][1])-240))
137     #             print str(int(track_box[0][0])-320) + " " + str(int(track_box[0]
138 [1])-240)
139
140         ##如果捕捉到的颜色区域高度小于等于1，则重新开始查找
141         if track_box[1][1] <= 1:
142             self.tracking_state = 0
143             self.start()
144             print(track_box)
145             print("=====")
146         else:
147             if self.show_backproj:
148                 #将查询到的背景值赋给vis
149                 vis[:] = prob[...,np.newaxis]
150             try:
151                 #画为椭圆
152                 cv2.ellipse(vis, track_box, (0, 0, 255), 2)
153                 #             print track_box
154                 # 打印中心的x, y坐标; 宽, 高; 旋转角度
155                 a = str(track_box[0][0])+" "+str(track_box[0][1])+"
156 "+str(round(track_box[1][0],2))\
157                 + " "+str(round(track_box[1][1],2))+"
158 "+str(round(track_box[2],2))+"\r\n"
159                 print(a)
160                 #             self.ser.write(a)
161             except: print(track_box)
162
163         cv2.imshow('camshift', vis)
164
165         ch = 0xFF & cv2.waitKey(5)
166         if ch == 27:
167             break
168         if ch == ord('b'):
169             #切换显示模式

```

```

163         self.show_backproj = not self.show_backproj
164         if ch == ord('r'):
165             #重新捕捉颜色区域
166             self.tracking_state = 0
167             self.start()
168         cv2.destroyAllWindows()
169
170
171 if __name__ == '__main__':
172     import sys
173     try: color = sys.argv[1]
174     except: color = 1
175     print(__doc__)
176     a = App(color)
177     a.run()
178

```

注意如果想要捕捉其他颜色的区域，需要加入读取图片后，更改tracking_state = 1。

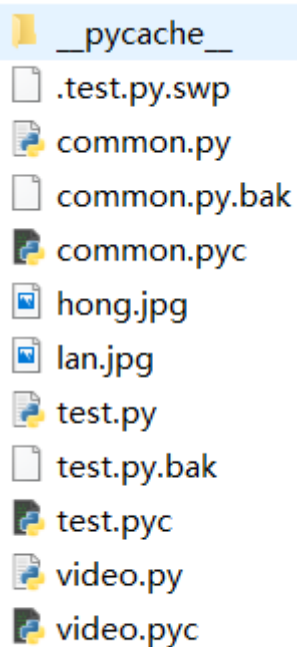
以及涉及到运行程序时输入参数应将其转换为整形参数。

```

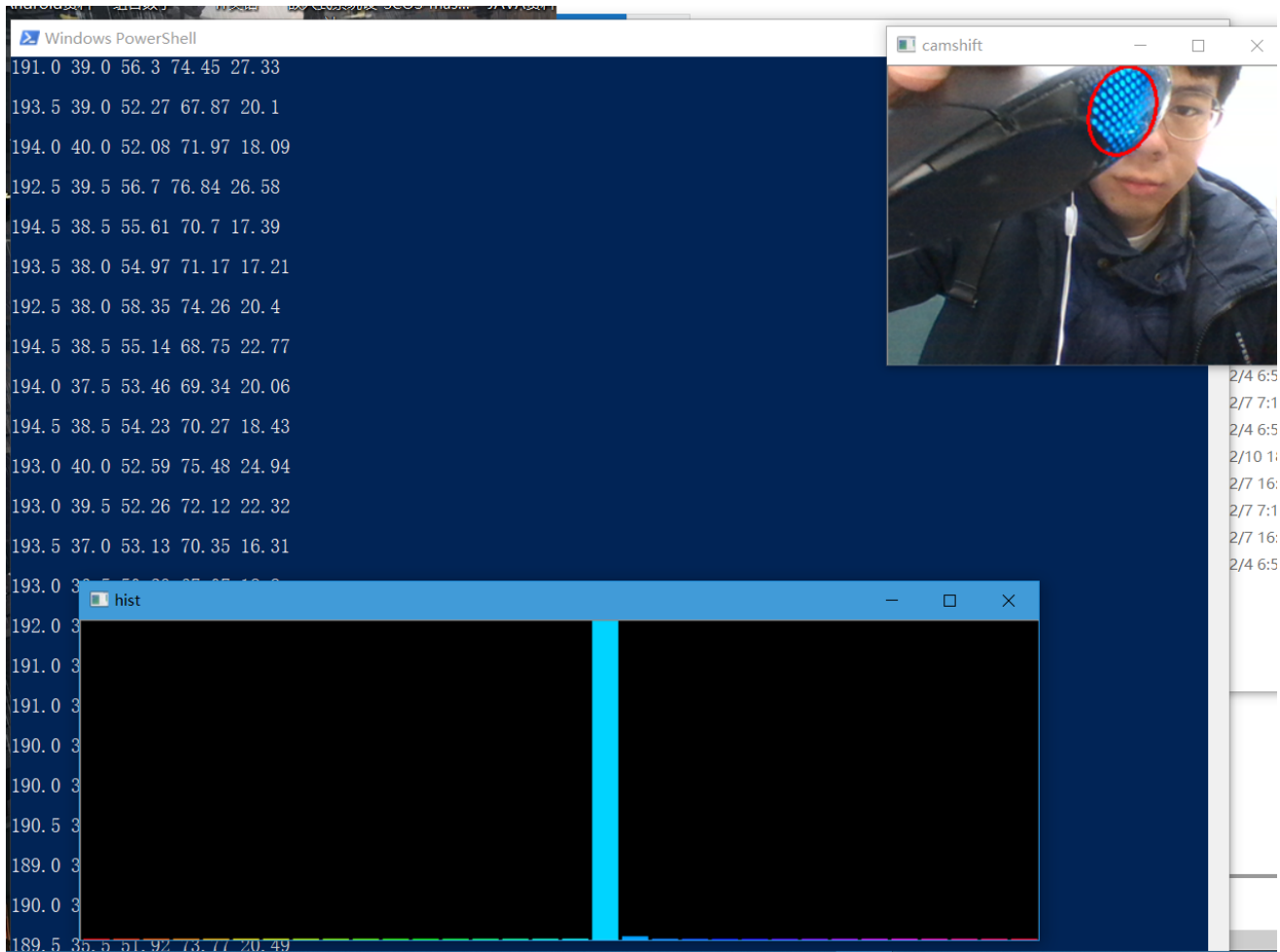
1 | int(color)

```

编译后，文件为



运行效果如图所示



191.0 39.0 56.3 74.45 27.33
193.5 39.0 52.27 67.87 20.1
194.0 40.0 52.08 71.97 18.09
192.5 39.5 56.7 76.84 26.58
194.5 38.5 55.61 70.7 17.39
193.5 38.0 54.97 71.17 17.21
192.5 38.0 58.35 74.26 20.4
194.5 38.5 55.14 68.75 22.77
194.0 37.5 53.46 69.34 20.06
194.5 38.5 54.23 70.27 18.43
193.0 40.0 52.59 75.48 24.94
193.0 39.5 52.26 72.12 22.32
193.5 37.0 53.13 70.35 16.31

193.0 3
192.0 3
191.0 3
191.0 3
190.0 3
190.0 3
190.5 3
189.0 3
190.0 3
189.5 35.5 51.92 73.77 20.49

2/4 6:5
2/7 7:1
2/4 6:5
2/10 1:
2/7 16:
2/7 7:1
2/7 16:
2/4 6:5