People Counter 9 – Counting

Last chapter! Again, sorry for delays..

Last time I showed you how to follow an object's movement, although really it's only saving a list of that object's previous coordinates. Now, we have to take a look at that list and determine if the object's moving up or down in the image.

To do this I'll first create two imaginary lines that'll indicate when to evaluate the object's direction (line_up, line_down).

I also set two limiting lines to tell my code when to stop tracking an object (up_limit, down_limit).

I also use two methods on the Person class, going_UP(a,b) and going_DOWN(a,b). Both receive line_down and line_up and return true is they evaluate if the object has crossed line_up or line_down in the correct direction. If so, then a counter is incremented.. and we're counting people.

Also, the Person class has a State attribute which is used to know when the object is outside the counting limits of the image and release allocated memory.

Here's code:

```
##Contador de personas
##Federico Mejia
import numpy as np
import cv2
import Person
import time

#Contadores de entrada y salida
cnt_up   = 0
cnt_down = 0

#Fuente de video
#cap = cv2.VideoCapture(0)
cap = cv2.VideoCapture('peopleCounter.avi')

#Propiedades del video
##cap.set(3,160) #Width
##cap.set(4,120) #Height

#Imprime las propiedades de captura a consola
for i in range(19):
    print i, cap.get(i)

w = cap.get(3)
h = cap.get(4)
frameArea = h*w
areaTH = frameArea/250
print 'Area Threshold', areaTH

#Lineas de entrada/salida
line_up = int(2*(h/5))
line_down   = int(3*(h/5))

up_limit =   int(1*(h/5))
down_limit = int(4*(h/5))

print "Red line y:",str(line_down)
print "Blue line y:", str(line_up)
line_down_color = (255,0,0)
line_up_color = (0,0,255)
pt1 =  [0, line_down];
pt2 =  [w, line_down];
pts_L1 = np.array([pt1,pt2], np.int32)
pts_L1 = pts_L1.reshape((-1,1,2))
pt3 =  [0, line_up];
pt4 =  [w, line_up];
pts_L2 = np.array([pt3,pt4], np.int32)
pts_L2 = pts_L2.reshape((-1,1,2))

pt5 =  [0, up_limit];
pt6 =  [w, up_limit];
pts_L3 = np.array([pt5,pt6], np.int32)
pts_L3 = pts_L3.reshape((-1,1,2))
pt7 =  [0, down_limit];
pt8 =  [w, down_limit];
pts_L4 = np.array([pt7,pt8], np.int32)
pts_L4 = pts_L4.reshape((-1,1,2))

#Substractor de fondo
fgbg = cv2.createBackgroundSubtractorMOG2(detectShadows = True)

#Elementos estructurantes para filtros morfoogicos
kernelOp = np.ones((3,3),np.uint8)
kernelOp2 = np.ones((5,5),np.uint8)
kernelCl = np.ones((11,11),np.uint8)

#Variables
font = cv2.FONT_HERSHEY_SIMPLEX
persons = []
max_p_age = 5
pid = 1

while(cap.isOpened()):
##for image in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    #Lee una imagen de la fuente de video
    ret, frame = cap.read()
##    frame = image.array

    for i in persons:
        i.age_one() #age every person one frame
    ########################
    #   PRE-PROCESAMIENTO   #
    ########################

    #Aplica substraccion de fondo
    fgmask = fgbg.apply(frame)
    fgmask2 = fgbg.apply(frame)

    #Binariazcion para eliminar sombras (color gris)
    try:
        ret,imBin= cv2.threshold(fgmask,200,255,cv2.THRESH_BINARY)
        ret,imBin2 = cv2.threshold(fgmask2,200,255,cv2.THRESH_BINARY)
        #Opening (erode->dilate) para quitar ruido.
        mask = cv2.morphologyEx(imBin, cv2.MORPH_OPEN, kernelOp)
        mask2 = cv2.morphologyEx(imBin2, cv2.MORPH_OPEN, kernelOp)
        #Closing (dilate -> erode) para juntar regiones blancas.
        mask =  cv2.morphologyEx(mask , cv2.MORPH_CLOSE, kernelCl)
        mask2 = cv2.morphologyEx(mask2, cv2.MORPH_CLOSE, kernelCl)
    except:
        print('EOF')
        print 'UP:',cnt_up
        print 'DOWN:',cnt_down
        break
    #################
    #   CONTORNOS   #
    #################

    # RETR_EXTERNAL returns only extreme outer flags. All child contours are left behind.
    _, contours0, hierarchy = cv2.findContours(mask2,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    for cnt in contours0:
        area = cv2.contourArea(cnt)
        if area > areaTH:
            #################
            #   TRACKING    #
            #################

            #Falta agregar condiciones para multipersonas, salidas y entradas de pantalla.

            M = cv2.moments(cnt)
            cx = int(M['m10']/M['m00'])
            cy = int(M['m01']/M['m00'])
            x,y,w,h = cv2.boundingRect(cnt)

            new = True
            if cy in range(up_limit,down_limit):
                for i in persons:
                    if abs(cx-i.getX()) <= w and abs(cy-i.getY()) <= h:
                        # el objeto esta cerca de uno que ya se detecto antes
                        new = False
                        i.updateCoords(cx,cy)   #actualiza coordenadas en el objeto and resets age
                        if i.going_UP(line_down,line_up) == True:
                            cnt_up += 1;
                            print "ID:",i.getId(),'crossed going up at',time.strftime("%c")
                        elif i.going_DOWN(line_down,line_up) == True:
                            cnt_down += 1;
                            print "ID:",i.getId(),'crossed going down at',time.strftime("%c")
                        break
                    if i.getState() == '1':
                        if i.getDir() == 'down' and i.getY() > down_limit:
                            i.setDone()
                        elif i.getDir() == 'up' and i.getY() < up_limit:
                            i.setDone()
                    if i.timedOut():
                        #sacar i de la lista persons
                        index = persons.index(i)
                        persons.pop(index)
                        del i     #liberar la memoria de i
                if new == True:
                    p = Person.MyPerson(pid,cx,cy, max_p_age)
                    persons.append(p)
                    pid += 1
            #################
            #   DIBUJOS     #
            #################
            cv2.circle(frame,(cx,cy), 5, (0,0,255), -1)
            img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
            #cv2.drawContours(frame, cnt, -1, (0,255,0), 3)

    #END for cnt in contours0

    ########################
    # DIBUJAR TRAYECTORIAS  #
    ########################
    for i in persons:
##        if len(i.getTracks()) >= 2:
##            pts = np.array(i.getTracks(), np.int32)
##            pts = pts.reshape((-1,1,2))
##            frame = cv2.polylines(frame,[pts],False,i.getRGB())
##        if i.getId() == 9:
##            print str(i.getX()), ',', str(i.getY())
        cv2.putText(frame, str(i.getId()),(i.getX(),i.getY()),font,0.3,i.getRGB(),1,cv2.LINE_AA)

    #################
    #   IMAGANES    #
    #################
    str_up = 'UP: '+ str(cnt_up)
    str_down = 'DOWN: '+ str(cnt_down)
    frame = cv2.polylines(frame,[pts_L1],False,line_down_color,thickness=2)
```

```
        cv2.putText(frame, str_up ,(10,40),font,0.5,(255,255,255),2,cv2.LINE_AA)
        cv2.putText(frame, str_up ,(10,40),font,0.5,(0,0,255),1,cv2.LINE_AA)
        cv2.putText(frame, str_down ,(10,90),font,0.5,(255,255,255),2,cv2.LINE_AA)
        cv2.putText(frame, str_down ,(10,90),font,0.5,(255,0,0),1,cv2.LINE_AA)

        cv2.imshow('Frame',frame)
        #cv2.imshow('Mask',mask)

        #preisonar ESC para salir
        k = cv2.waitKey(30) & 0xff
        if k == 27:
            break
    #END while(cap.isOpened())

    ################
    #   LIMPIEZA   #
    ################
    cap.release()
    cv2.destroyAllWindows()
```

(You'll need the Person.py file located in another part of this tutorial to run the code)

As you may see, the code is not very precise when counting. You can always add more lines to make it more exact, but more processing will be needed, such as here:

Opencv Python - People Counter 2

Please let me know if you have any suggestions on how to count more efficiently.

—

I hope you've enjoyed this tutorial, even if it took me this much to finish. I'll be very glad to know if you use this code, or your own, in some application. ☺

I'll try to answer any further questions through the comment section.

Fede / January 26, 2017 / People counter

4 thoughts on "People Counter 9 – Counting"

dbispo
February 27, 2017 at 10:59 am

I receive the error in last code:

OpenCV Error: Assertion failed (The data should normally be NULL!) in NumpyAllocator::allocate, file C:\builds\master_PackSlaveAddon-win64-vc12-static\opencv\modules\python\src2\cv2.cpp, line 163
Traceback (most recent call last):

File "C:/CounterPeople.py", line 86, in
fgmask = fgbg.apply(frame)
cv2.error: C:\builds\master_PackSlaveAddon-win64-vc12-static\opencv\modules\python\src2\cv2.cpp:163: error: (-215) The data should normally be NULL! in function NumpyAllocator::allocate
Log in to Reply

andrecz
April 10, 2017 at 4:45 pm

Hola Federico,
soy un aficionado al python y opencv. Estoy intentando usar VideoStream de la libreria imutils para poder usar la picamera de raspberry pi con tu codigo. No lo habras probado? O no sabes de que forma se podría integrar. Lo estoy probando, pero dado que no soy programador voy un poco perdido, y puede ser que la solucion sea facil y yo no la veo. Si no te importára darme un consejo te lo agradecería y si no tampoco pasa nada, gracias de todas formas por compartir tu codigo.
Andres
Log in to Reply

Fede
April 17, 2017 at 9:23 pm

Hola, sí lo he hecho... algo asi:

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import numpy as np
import cv2
import Person
import time, datetime
import signal,sys
.
.
.
camera = PiCamera()
camera.resolution = (160,120)
camera.framerate = 5
rawCapture = PiRGBArray(camera, size=(160,120))
time.sleep(0.1)
.
.
.
for image in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
frame = image.array
.
.
.
```

Log in to Reply

guilherme
June 1, 2017 at 1:20 pm

Congratulations very good your tutorial for this accountant. I did everything as explained and I'm running on architecture arm.
I would like to know if you already have a

```
        ##################
         # TRACKING #
         ##################

         # Fail to add conditions for multipersons, outputs and screen inputs.
```

That in my case I have to count multiple people. Coming in and out.
If you already have it you can send me an email or post here and tell you how to do it.
Thank you very much.

Enhorabuena muy bien tu tutorial para ese contador. Lo hice todo según lo explicado y estoy rodando en arquitectura arm.
Me gustaría saber si ya tiene una logica para

```
        ######################
         # TRACKING #
         ######################

         #Falta agregar condiciones para multipersonas, salidas y entradas de pantalla.
```

En mi caso tengo que contar a varias personas. Entrando y saliendo.
Si ya posee puede mandarme un e-mail o publicar aquí y avisar de cómo hacerlo.
Muchas gracias.
guilherme
guilhermelourenco2005@gmail.com
Log in to Reply